

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 04-291659

(43)Date of publication of application : 15.10.1992

(51)Int.Cl.

G06F 15/16

(21)Application number : 03-056467

(71)Applicant : HITACHI LTD

(22)Date of filing : 20.03.1991

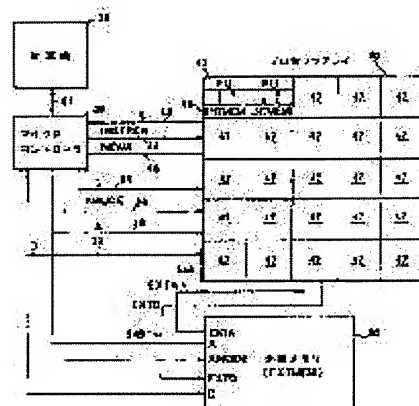
(72)Inventor : MARIUSU HANKU  
IWASAKI KAZUHIKO  
SATO YUJI

## (54) PARALLEL COMPUTER SYSTEM AND ITS OPERATION METHOD

### (57)Abstract:

**PURPOSE:** To execute a different instruction in each processor group out of plural processors to which a single instruction is applied.

**CONSTITUTION:** A microcontroller 30 outputs a primary instruction to a bus 31, outputs a coincidence word for specifying a processor zone to an address bus 39 and outputs a signal AMODE for specifying an operation mode to a bus 34. Each processor has a circuit for judging which processor zone includes itself and a circuit for forming a secondary instruction by correcting an inputted primary instruction when a mode signal is an instruction correcting mode. The secondary instruction can be set up also as a NO OPERATION instruction. Since the secondary instruction can be generated from the circuit of each processor, a different instruction can be executed in each processor zone.



## LEGAL STATUS

[Date of request for examination]

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

(19) 日本国特許庁 ( J P )

(12) 公 開 特 許 公 報 ( A )

(11) 特許出願公開番号

特開平4-291659

(43) 公開日 平成 4 年 (1992) 10 月 15 日

(51) Int.Cl.<sup>5</sup>

G 0 6 F 15/16

識別記号

庁内整理番号

3 9 0 T 9190-5L

F I

技術表示箇所

審査請求 未請求 請求項の数16(全 40 頁)

(21) 出願番号

特願平3-56467

(22) 出願日

平成 3 年 (1991) 3 月 20 日

(71) 出願人 000005108

株式会社日立製作所

東京都千代田区神田駿河台四丁目 6 番地

(72) 発明者 マリウス・ハंक

東京都国分寺市東恋ヶ窪 1 丁目 280 番地

株式会社日立製作所中央研究所内

(72) 発明者 岩崎 一彦

東京都国分寺市東恋ヶ窪 1 丁目 280 番地

株式会社日立製作所中央研究所内

(72) 発明者 佐藤 裕二

東京都国分寺市東恋ヶ窪 1 丁目 280 番地

株式会社日立製作所中央研究所内

(74) 代理人 弁理士 小川 勝男

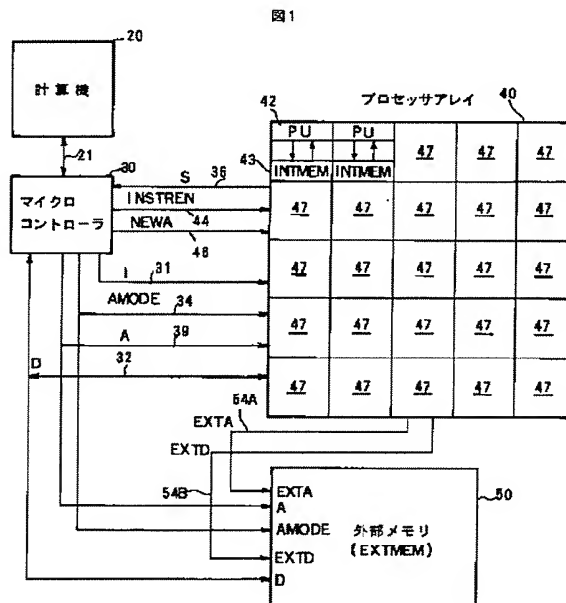
(54) 【発明の名称】 並列コンピュータシステムおよびその動作方法

(57) 【要約】

【目的】 単一命令が与えられる複数のプロセッサにおいて、プロセッサグループごとに異なる命令を実行可能にする。

【構成】 マイクロコントローラ 30 より一次命令をバス 31 に出力し、アドレスバス 39 にプロセッサゾーンを指定するための一致ワードを出力し、バス 34 に動作モードを指定する信号 AMODE を出力する。各プロセッサは、自己がどのプロセッサゾーンに属するかを判別する回路と、このモード信号が命令修正モードのとき入力された一次命令を修正した二次命令を生成する回路を有する。この二次命令は、No operation 命令とすることもできる。

【効果】 各プロセッサ内の回路により、二次命令を発生できるので、プロセッサゾーンごとに異なる命令を実行可能になる。



1

## 【特許請求の範囲】

【請求項1】複数のプロセッサのアレーからなる並列コンピュータシステムにおいて、各プロセッサに対し同一の一次命令を送り、各プロセッサに対し前記プロセッサアレーのプロセッサのうち選択されたプロセッサのグループを特定する第1の信号を送り、各プロセッサにおいてそのプロセッサが前記選択されたプロセッサのグループに属するかどうかを検出し、該選択されたプロセッサグループに属すると検出された複数のプロセッサの各々において、該受け取った一次命令から、その命令の代

わりに実行すべき2次命令を生成するというステップからなる、アレーのプロセッサを動作させる方法。

【請求項2】各プロセッサが該一次命令及び該第1の信号を並列に受け取る、請求項1記載の方法。

【請求項3】更に、プロセッサのそれぞれに対し、選択されたプロセッサのグループに属するプロセッサによって生成されるべき命令に関連する第2の信号を送り、該二次命令の生成は、受け取られた一次命令及び前記第2の信号に基づいて行う請求項1記載の方法。

【請求項4】各プロセッサが、共通のバスを通して前記第1の信号及び前記第2の信号を並列に受け取る、請求項3記載の方法。

【請求項5】前記2次命令の生成は、受け取った命令の一部を前記第2の信号に依存して定まる新しい情報によって置き換えることを含む請求項3記載の方法。

【請求項6】第1の信号が複数のプロセッサグループを特定する情報を含み、そして第2の信号はそれぞれ、該複数のプロセッサグループの一つに属するプロセッサによって生成される複数の命令に関連し、該二次命令の生成は、プロセッサが属するプロセッサグループ、及び対応する第2の信号に基づいている請求項3に記載の方法。

【請求項7】前記第1の信号及び前記第2の信号が、アドレス情報による時分割の方法により、プロセッサアレーをコントロールするアドレスバスへ送られる、請求項1、2、3、4、5及び6のいずれか1つの項に記載の方法。

【請求項8】プロセッサのアレーからなる並列コンピュータシステムにおいて、プロセッサアレーを多数の第1のプロセッサグループに分割し、第1のプロセッサグループの全てのプロセッサに対し同一の初期命令を送り、第1のプロセッサグループの各プロセッサに対し前記プロセッサグループのプロセッサの中の多数の第2のプロセッサグループを特定する共通の第1の信号を並列に送り、第1のプロセッサグループの中の各プロセッサに対し、前記第1のプロセッサグループのそれぞれの第2のプロセッサグループに属するプロセッサによって生成される命令に関連する共通の第2の信号を並列に送り、受け取った命令の代わりに実行すべき2次命令を各プロセッサにおいてローカルに生成し、前記生成が、受け

2

取った初期命令、第2のプロセッサグループに属するプロセッサであることを検出した結果、及び前記第2のプロセッサグループに対応する前記第2の信号、に基づいているというステップからなるアレーのプロセッサを動作させる方法。

【請求項9】コントロール回路手段から同一の初期命令を受け取るプロセッサ手段のアレーからなる並列コンピュータシステムにおいて、前記プロセッサアレー手段のプロセッサの中から選択されたプロセッサのグループを特定する第1の信号ライン手段と、各プロセッサの中に含まれ、そのプロセッサが選択されたプロセッサのグループに属するかどうかを検出するために、受け取った第1の信号に応答する手段と、受け取った命令の代わりに実行すべき2次命令を、各プロセッサの位置でローカルに生成するために各プロセッサの中に含まれる手段であって、前記生成が、受け取った初期命令及びそのプロセッサが前記プロセッサグループに属することを検出した結果に基づいている。という手段からなり、各プロセッサにおいて2次命令を生成し、及び実行する装置。

【請求項10】各プロセッサが、コントロール回路手段から初期命令及び第1の信号を並列に受け取る、請求項9記載の装置。

【請求項11】更に、選択されたプロセッサのグループに属するプロセッサによって生成される命令に関連する第2の信号ライン手段と、各プロセッサに含まれ、受け取った命令の代わりに実行すべき2次命令を各プロセッサにおいてローカルに生成するための手段であって、前記生成が更に、受け取った初期命令及びそのプロセッサが選択されたプロセッサグループに属することを検出した結果に加え、前記第2の信号に基づいてなされる、という手段からなる請求項9記載の装置。

【請求項12】各プロセッサが、共通バス手段を通して前記第1の信号及び前記第2の信号を並列に受け取る、請求項11に記載の装置。

【請求項13】前記ローカルな2次命令の生成には、受け取った命令を新しい情報によって置き換えることを含み、それが、プロセッサアレー手段の選択されたプロセッサによって実行すべき新しい命令を生成するために、前記第2の信号に依存している請求項11に記載の装置。

【請求項14】第1の信号が複数のプロセッサグループを特定し、第2の信号が各プロセッサグループに属するプロセッサによって生成される命令にそれぞれ関連し、各プロセッサにおける2次命令のローカルな生成が、受け取った命令、そのプロセッサが属していると検出されたプロセッサグループ、及び対応する第2の信号に基づいている請求項11に記載の装置。

【請求項15】前記第1の信号及び前記第2の信号が、アドレス情報による時分割の方法により、プロセッサア

3

レーをコントロールするアドレスバスへ送られる、請求項9, 10, 11, 12, 13, 及び14のいずれか1つの項に記載の装置。

【請求項16】同一の初期命令が第1のプロセッサグループのすべてのプロセッサに送られる複数の第1のプロセッサグループに分割された、プロセッサ手段のアレーから成る並列コンピュータシステムにおいて、前記第1のプロセッサグループのプロセッサの中の多数の第2のプロセッサグループを特定するとともに、第1プロセッサグループのそれぞれのプロセッサに共通に送られる第1の信号手段と、前記第1のプロセッサグループの各第2のプロセッサグループに属するプロセッサによって生成される命令に関連し、第1のプロセッサグループ内の各プロセッサに共通に送られる第2の信号手段と、各プロセッサに含まれ、受け取った命令の代わりに実行すべき2次的な命令を各プロセッサにおいてローカルに生成するための手段であって、前記生成が、第1のプロセッサグループによって受け取られる初期命令、そのプロセッサが第2のプロセッサグループに属することが検出された結果、及び前記第2のプロセッサグループに対応する前記第2の信号に基づいてなされる、という手段からなる、ことを特徴とするアレーのプロセッサを動作させる装置。

【発明の詳細な説明】

【0001】

【産業上の利用分野】本発明は複数のプロセッサからなる並列プロセッサに係り、とくに一度にいくつもの命令を同時に生成しかつ実行するために単一命令流(single instruction stream)を用いる並列コンピュータに好適な並列プロセッサに関連する。

【0002】

【従来の技術】従来のコンピュータシステムは、プログラムメモリからの単一命令流の中から一度に一つ与えられる命令を実行する単一のプロセッサ(中央処理装置-CPU)から構成されていた。これは逐次型単一プロセッサ・アーキテクチャ(フォン・ノイマン型)として知られている。

【0003】このようなアプローチはつい最近までは成功していたが、計算に対するニーズが絶え間なく増大したために、コンピュータ設計者によって新しいアーキテクチャの実験が行なわれてきている。これらの実験では、単にプロセッサ自体の性能を改善するだけでなく、いわゆる高度並列アーキテクチャまたは大規模並列アーキテクチャと呼ばれているように非常に多くのプロセッサを結合することによって、主として計算速度及びスループットの向上が追求されている。このようなアーキテクチャでは、計算の仕事は、命令を同時に実行する多数の処理エレメントに配分される。このようなアプローチは非常に望ましいものであると認められている。これについては、例えば、シー・ミードとエル・コンウェイ

4

(C. Mead and L. Conway)による“VLSIシステム入門”(Introduction to VLSI Systems) (Addison Wesley, 1980, pp264-265 and 273)及びそこで引用されている参考文献、さらにW. ヒリス(W. Hillis)による“コネクション マシン”(The Connection Machine) (The MIT Press, 1985, pp3-6)及びそこで引用されている参考文献を参照。

【0004】さらに、より利用価値の高いアーキテクチャとしては、MIMD (Multiple-Instruction Multiple-Data: 複数命令複数データ) が従来からの技術としてある(例えば、ホワングとブリッグス(Hwang & Briggs)による“コンピュータアーキテクチャと並列処理”(Computer Architecture and Parallel Processing) (McGraw-Hill, 1984, p.26参照)。これはいくつかの命令の流れが複数のプロセッサエレメントへ送られ、このエレメントは同時にいくつかの命令を実行するようプログラムされている。これにより遊んでいるプロセッサの数をかなり減らすことができ、並行性(Concurrency)が向上する。また、潜在的に高い計算速度が達成可能である。

【0005】しかしながらMIMDアーキテクチャについては、異なる命令相互接続バスを、プロセッサアレーの中の、異なる複数組のプロセッサに配線することに関する問題に直面することになるので、コンピュータ設計者は、いまだに魅力的とは感じていない。また、プロセッサをコントロールするマイクロコントローラは、より多くの命令信号を同時に生成しなければならないので、設計がより複雑となる。このため、MIMDアーキテクチャの大規模の並列コンピュータは、現段階ではきわめて複雑な装置とならざるを得ない。

【0006】このため、最近まで使用されていた大規模並列コンピュータの多くは、いわゆるSIMD (Single-Instruction Multi-Data: 単一命令複数データ) と呼ばれる種類に属している。このSIMDアーキテクチャにおいては、非常に多くのプロセッサをコントロールするという問題は、プロセッサアレーを構成する全てのプロセッサに同一の命令を送ることによって単純なものとされている。同時に、場合によって異なるローカルなデータの組(初期データ及び中間結果)がそれぞれのプロセッサエレメントにおいて存在することが許されている。この種の並列コンピュータはよく知られている。例えば、前述のホワングとブリッグス(Hwang & Briggs)による“コンピュータアーキテクチャと並列処理”(Computer Architecture and Parallel Processing) (McGraw-Hill, 1984, p.24)を参照。またThinking Machine Inc. の Connection Machine CM-1 (前述の文献“The Connection Machine”, The MIT Press, 1985, p.21参照)及びMasPar の MP-1 (ジェイ・アール・ニコルズ(J. R. Nickols)による“The Design of the MasPar MP-1: A Cost-Effectively-Parallel Computer”, Pro

c. Compcon Spring 1990, pp.25-28参照) などはこの範疇に含まれる。

【0007】図12は、命令の単一の流れによって駆動される並列処理集積回路のアレーを用いた従来の並列コンピュータシステムを図式的に示すもので、ダブリュー・ヒルス(W. Daniel Hills)他による「並列プロセッサ/メモリ回路」という名称の合衆国特許4,709,327(1987年11月24日発行)に記載されているものである。この図において、並列コンピュータシステムは、汎用コンピュータ710、マイクロコントローラ720、及びプロセッサアレー730から構成されている。プロセッサアレーは、集積回路として与えられた多数のプロセッサ735からなり、それぞれが接続用のネットワークによって相互に接続されている。コンピュータ710は高級言語でプログラミングを行うホストとして使用され、双方向のバスを介してマイクロコントローラ720をコントロールする。このマイクロコントローラ720はバス722を介してプロセッサアレー730に対して命令の単一の流れを与える。同時にこのマイクロコントローラ720は、アドレスバス724によってプロセッサアレーにアドレス情報を与える。このアレーの状態は単一のライン726を介してマイクロコントローラ720によってサンプルされる。このプロセッサアレーの中の全てのプロセッサが、マイクロコントローラ720から送られて来る同一の命令を実行する。これは典型的なSIMDアーキテクチャの例である。

【0008】SIMD並列アーキテクチャはプログラミングの簡単さやコントローラの設計という点で有利であると共に、同一の組のワイヤー(命令ビットを伝達する)がアレーの中の全てのプロセッサに接続されるので、マイクロコントローラからプロセッサアレーへの配線の点についても便利である。このようなプロセッサエレメントは、しばしばマイクロコントローラによって駆動されるプロセッサアレーとして構成される。

【0009】

【発明が解決しようとする課題】SIMDアーキテクチャの不利の点の一つは、命令流内の現在の命令を実行しないようにすべき一部の複数のプロセッサは(これは、アルゴリズムのアレーへのマッピングによって定まる)、一時的にその命令に回答しない状態にしなければならないことである。通常、この目的のために現在の命令が伝えられるのに先立ってそれらの一部のプロセッサの各々内のマスクレジスタに、そのプロセッサをイネーブルにしたりディセーブルにする情報を選択的にロードしているが、この手続はかなりのオーバーヘッドがある。

【0010】さらに、現在の命令を実行しないプロセッサが増大すると、リソースの有効利用の点で、望ましくない。

【0011】本発明の目的は、単一の命令流を供給され

るプロセッサアレーの一部のプロセッサをより簡単な方法で、非応答状態とすることのできる方法およびそのための並列コンピュータシステムを提供することにある。

【0012】本発明の他の目的は単一の命令流を供給されるプロセッサアレーの一部のプロセッサの各々により異なる命令を実行させられる、並列プロセッサシステムを提供することにある。

【0013】本発明のさらに他の目的は、複数の命令流の一つの命令流がそれぞれ供給される複数のプロセッサグループからなるいわゆるMIMD形式の並列プロセッサであって、各グループ内のプロセッサの一部のプロセッサを、そのグループに供給される命令流の各命令に回答しないかあるいはその命令とは異なる命令を実行する状態に、容易に変更する方法とそのための並列プロセッサシステムを提供することにある。

【0014】

【課題を解決するための手段】プロセッサアレー中の各プロセッサにおいてマイクロコントローラから送られるたった一つの一次命令から、別の命令が生成でき、複数の命令を結果的に並列に実行でき、遊んでいる状態にあるプロセッサを少くできる。

【0015】また、一部のプロセッサの各々において、上記一次命令に回答しないようにできるので、このためのオーバーヘッドが少なくとすむ。

【0016】前述の、及びそれ以外の目的、特徴、発明の利点は、以下に続く、添付した図面に示される本発明の好ましい具体例の詳しい説明から明かとなるだろう。

【0017】

【実施例】本発明の実施例は、読者の便宜を考慮して以下のセクションに分けて説明する。

【0018】○システムの一般的な説明

○プロセッサ47の一般的な説明

○アドレス一致ユニット(ACU)52

○メモリアンターフェース(MEMIF)53

○アドレス一致回路81

○命令修正回路82

○変形例

○他の実施例

○システムの一般的な説明

図1は、単一の集積回路上に形成されたプロセッサアレーを用いた、本発明による並列コンピュータシステムを概略的に示したものである。これはSIMDアーキテクチャによるものである。図1に示すように、本発明は、コンピュータ20、マイクロコントローラ30、相互に接続されたプロセッサ47のプロセッサアレー40、及び外部メモリEXTMEM50からなるコンピュータシステムで実施されてよい。

【0019】ここには、アレー40内のプロセッサとして5行5列に配列されたプロセッサを示しているが、この数は一例にすぎない。

【0020】コンピュータ20は汎用ホストコンピュータ又はワークステーションである。このコンピュータ用のプログラムはユーザーの便宜を考慮して高級並列言語によってプログラミングされる。コントロールバス21は、コンピュータ20とマイクロコントローラ30とのインターフェース用として使用する。マイクロコントローラ30は、コンピュータ20から与えられる高級言語の命令の各々に応答して、プロセッサアレー40の駆動に使用される一連のマイクロ命令を生成する。マイクロコントローラ30は命令バス(I)31を介して、プロセッサアレー40内の全てのプロセッサ47に、生成した一連のマイクロ命令を順次与える。各マイクロ命令は個々のプロセッサによって同時に実行されるべき同一の演算を規定する。以下ではマイクロ命令を単に命令又は一次命令と並ぶことがある。双方向のデータバス(D)32は、アレー40内のすべてのプロセッサに接続され、個々のプロセッサへのデータの入力及びそこからデータの出力用に使用される。アドレスバスA39は、プロセッサアレー40に対してアドレス情報を与える。アドレスバスの構成についての詳しい説明は、後述する。ステータスバス36も、全てのプロセッサに接続され、プロセッサアレー40の個々のプロセッサの現在の状態を表わすステータス情報をマイクロコントローラ30に与えるのに使用される。このステータス情報は、たとえば、全てのプロセッサにおける演算の終了(算術、ロジック又はデータルーチング)をマイクロコントローラ30が検出するのに使用する。このステータス情報を利用するためには、マイクロコントローラ30によってプロセッサアレー40に、それぞれのプロセッサのステータスフラグの値に依存して異なる動作を指定する、条件付きのマイクロ命令を伝達し、この命令により、全てのプロセッサから、それぞれ内のステータスフラグを適当なタイミングでステータスバス36上に出力させる。こうして、各プロセッサのステータス情報をマイクロコントローラ36に通知する。マイクロコントローラ30は、全てのプロセッサからのステータスフラグの値に依存して次のマイクロ命令を選択する。我々の好ましい具体例では、ステータスバス36は、プロセッサアレー40に対し共通に設けた1ビットのバスであり、アレー40の中の全てのプロセッサのそれぞれの1ビットのステータスフラグがワイヤードANDされ、その結果信号Sがステータスバス36に出力される。

【0021】プロセッサアレー40によって実行される並列プログラムに対する初期及び中間のデータは、各プロセッサの内部(ローカル)メモリ(INTMEM)43及び各プロセッサからアクセス可能な、全てのプロセッサに共通に設けた外部メモリ(EXTMEM)50に保持される。外部メモリの必要性は、各プロセッサ47は好ましくは集積回路とされるので、その内部ではRAM(Random-Access Memory)及びROM(Read-OnlyMem

ory)として使用できる空間が限られているという事実に基づいて生じる。また、このように集積回路技術を用いると、外部に低コストで大きなメモリを設ける方がより簡単となる。この外部の大きな実体的なメモリは、例えば仮想メモリの配置で、全てのプロセッサ47によって共有され、これにより外部メモリのセグメントはそれぞれのプロセッサに割り当てられる。できればこれらのセグメントは、いくつかのプロセッサから外部メモリの同一セグメントへのアクセスの衝突を避けるために、互いにオーバーラップしていないことが望ましい。

【0022】マイクロコントローラ30は読み出し/書き込み動作を行うときに、適当なアドレスの配置によって、これらのプロセッサの内部レジスタ、内部メモリ43、又は外部メモリ50を選択することができる。

【0023】外部メモリ50はデュアルポートタイプである。これにより、読み出し及び書き込みの動作は、マイクロコントローラ30と、プロセッサアレー40の両方から行うことができる。このようなことを可能とするために、データバス(D)32もアドレスバス(A)39も外部メモリ50に対しても使用できるようにする。同時にプロセッサアレー40は、データを外部メモリ50に与えるための独自のデータバス(EXTD)54B、及び独自のアドレスバス(EXTA)54Aを有している。マイクロコントローラ30とプロセッサアレー40の両方から同時に外部メモリ50へ書き込みを行うという危険をなくすために、ステータスバス36をサンプリングすることによって、マイクロコントローラ30はプロセッサアレー40が現在実行している動作の特性及び状態を知り、外部メモリ50へのアクセスの可否を判断するようになっている。

【0024】このようなSIMD(Single-Instruction Multiple-Data)アーキテクチャでの並列コンピュータシステムでは単一命令流がプロセッサアレー40の中の全てのプロセッサに伝達されるので全てのプロセッサ47に実行させるための命令を与えるのに、プロセッサアレー40の中の全てのプロセッサ47にたった一組のワイヤ(命令バスI)だけが接続されているので、構成が簡単になることが保証される。

【0025】従来技術では、しかしながら、プログラマーには、二つの異なる命令を同時に実行するようにプロセッサアレー40の中の異なる二つのプロセッサ47をプログラミングするという選択の余地はない。選択できるのは、伝達した命令を実行するか、又は何も実行しない(動作が行われない)かだけである。しかも、それぞれのプロセッサが現在の命令を実行するか否かは、この命令の発行前に、各プロセッサの命令マスクレジスタ(図示せず)に、1(そのプロセッサでの命令の実行をイネーブルにする)又は0(そのプロセッサでの命令の実行をディセーブルにする、又は動作を行わない)の値を書き込む必要がある。この従来技術ではこのような動



作モードの指定のためのオーバーヘッドが大きくなり、命令の実行の際の並行性が減少し、従って全体のスループット（単位時間当りに実行される命令の数）が小さくなる。

【0026】本実施例ではこの点を改善し、SIMDの構成の並列プロセッサシステムにMIMDの性能と柔軟性を与えるために、システムのアーキテクチャに対し以下のような独創的な変更を行った。すなわち、少なくとも一部のプロセッサの各々は、マイクロコントローラ30から伝達される（以下これを一次命令と呼ぶことがある）から別の命令（以下これを2次命令又は導出命令と呼ぶ）をローカルに生成し実行するように、再構成される。

【0027】この2次命令は、プロセッサによって異なりうる。また、この2次命令が何の操作も実行しない命令（NOP命令）のこともある。この再構成は、一つの具体例では、プロセッサアレー40の中の個々のプロセッサ47に記憶された、その位置に関する情報及びアドレスバス39によって伝えられるアドレス情報とAMODE信号線34により与えられ動作モード情報に基づいて決定される。プロセッサアレー40に対して、いくつかの動作モードが使用可能である。これらの動作モードの間の選択は、アドレスモードバス（AMODE）34によってなされる。アドレスサンプリング信号NEWA46は、マイクロコントローラ30によってプロセッサアレー40の中のアドレスラッチ（99（図6））に対して与えられる。

【0028】読者に対して我々の最終目的の考えを予め与えるために、提案した回路の一つの主要な動作モードを理解するのに役立つ図式表示として図11を使う。このモードでは、命令バス（I）31を介してマイクロコントローラ30から伝達された単一の一次命令から出発して、異なるプロセッサにおいて別々の命令を生成することが可能である。図11に与えられた例図では、簡単のために8×8個のプロセッサだけを持つプロセッサアレーを考える。401はプロセッサアレー40の中で同時に実行される命令のマップである。図11に示した場合において、オリジナルの命令（EX-OR）から、+（加算）475及び-（減算）477という二つの2次（導出）命令が生成される。図11を見ると分かるように、プロセッサアレー40は二つのゾーンに分けられ、それぞれのゾーンの中のプロセッサでは同一の導出命令を実行する（+又は-）。さらに図10は、本実施例の他の主要なモードで実行される命令のマップを示すもので、一部のプロセッサにおいて、EOR命令を実行させ、他はNOP命令を実行させることができる。

【0029】我々の考えに焦点を与えるために、このコンピュータシステムにおけるアドレス機構の動作の方法についていくつかの仮定を行う。まず、各プロセッサの内部メモリ（INTMEM）43及び外部メモリ（EX

TMEM）50に対するアドレス動作には、アドレスバス（A）39が独占的に使用されると仮定する。言い換えると、命令バス（I）31を介してプロセッサアレー40によって受け取られる一次命令には、これらのメモリをアドレスするための情報は含まれていない。この一次命令の唯一のアドレス能力は現在のALU又はI/Oに関する動作に関与する、各プロセッサの内部レジスタを選択することに限定される。

【0030】更に、アドレスバス（A）39はその全てのビットが全てのプロセッサ47に接続されている、すなわちプロセッサ47の外側ではアドレスのデコード動作は一切行われず、と仮定する。しかしこれは強制的な条件ではなく、変形例のセクションにおいて、プロセッサによって既にデコードされたアドレスが到達する場合にこの発明がどう適用されるかについて述べる。

【0031】上記の仮定を調べることによって、アドレスバス（A）39は全てのプロセッサに接続されているものの、プロセッサアレー40がメモリ参照をしない命令（全てのプロセッサに伝達される）を実行しているときには、実際上使用されていないということが理解される。このことは多くの命令について起こりうる。そしてこのアドレスバスが、実質的に時分割された多機能バスとなり得るという考えが導かれる。すなわち、アドレスバス（A）39の内容がメモリ参照と関係していない限りは、これをデコードするための適当な回路が存在するならば、他の目的のために自由に使用することができる。我々の発明においては、このバスを、元々は一つの命令だけを持つアーキテクチャ（SIMD）において後述するように、複数の命令の流れを生成するための情報を各プロセッサに与えるのに使用することにした。

【0032】プロセッサアレー40がメモリ参照を必要としない命令を実行しているタイムスロットではアドレスバス（A）39を直接インターラプトできるとはいえ、伝達される一次命令がこのようなメモリ参照を含んでいるときには、この様なことは不可能である。このようなときにはアドレスバス（A）39の内容は、どのメモリ位置をアクセスするかを決定するのに深く関連している。

【0033】このような二つの使用態様を実現可能とするために、アドレスバス（A）39上のアドレスは、内部メモリINTMEM43又は外部メモリEXTMEM50に分散される前にラッチされる（図6アドレスラッチ99）。このことは次のセクションで説明する。

【0034】図2は図1を理解するのに役立ついくつかの波形を示している。この図には、マイクロプロセッサ30によって生成される信号のタイミングダイアグラムが示してある。システム全体の同期を取るための基本となるクロック信号PHI1及びPHI2も同時に示す。これは互いに重ならない2相クロックで、図に示すように5乃至10MHzの範囲の周波数を持っている。



【0035】アドレスバス(A)39上の情報は6クロックサイクルの周期で更新される。これらの内容はプロセッサ内部レジスタ又は内部メモリ43又は外部メモリ50に対するアドレス情報として、または後述のアドレス一致回路によって使用されるプロセッサ選択のためのアドレス情報として、択一的に解釈される。

【0036】一次命令は命令バス(I)31によって伝達される。命令バス(I)31はクロックサイクルが7から12の時に有効な値(valid)となり、クロックサイクルが1から6及び13から18までが無関係な値(すなわち×印で示すドントケア)となる。なお、図2においてクロックサイクルは、信号の右端の位置で数える。この一次命令に対するストロブ信号としてマイクロプロセッサ30は、クロックサイクル8から11及び20から23において、ハイ(すなわち1。この文書全体を通して正論理であると仮定する)のときにアクティブとなる命令ストロブ信号(INSTREN)44を与える。

【0037】アドレス情報に対するストロブ信号としてマイクロコントローラ30は、図に示すようにクロックサイクルが5の間有効な値となるアドレスストロブ信号(NEWA)46を与える。

【0038】アドレスバスA上のアドレス情報の解釈を定義するアドレスモード信号(AMODE)34(後述するように、好ましい具体例では2ビットからなる)は、クロックサイクル7から12及び19から24において有効な値となる。

【0039】ここで提案されたマイクロコントローラ30の動作は、そのコントローラの実現の仕方には依存しない。このコントローラ30は、ROMやPLAを中心とする有限状態機械(オートマトン)のような通常の方法によっても実現することができる。このようにコントローラ30の実現の仕方自体は、本発明の目的ではない。

【0040】本実施例では、とくに断わらない限りマイ\*

表1

-----  
INSEL OUTSEL SHIFT IO F REGSEL INEN OEN ALU  
-----

【0045】ALU:現在の全命令に基づいて、プロセッサの算術論理回路(ALU)67(図4)によって実行される機能を選択する。

【0046】INSEL:ALU67上で演算を受けるべきオペランドの、入力セクタ66による選択動作を指示する。

【0047】OUTSEL:出力セクタ68によってバス101及び103の方へALU67の結果が転送される方向を指示する。

【0048】SHIFT:シフター65において行われるシフト動作の方向及び種類などを選択する。

\*クロコントローラ30から命令バス(I)29を通過して到達する一次命令はビット幅が29ビットで、ビット0が最下位ビット、ビット28が最上位ビットであると仮定する。同様に、アドレスバス(A)39のビット幅が10ビットであると仮定するが、本発明はこれらの数値以外にも適用されることは言うまでもない。

【0041】○プロセッサ47の一般的な説明

図3は、図1のプロセッサアレー40の1つのプロセッサ47の具体例を図式的に示したブロックダイアグラムである。このプロセッサ47は、他のプロセッサとともに単一の集積回路チップ上に設けられている。プロセッサアレー40はいくつかのプリント回路基板を含んでおり、このそれぞれは数千のプロセッサからなる。このようにして数万のプロセッサを含む大規模並列アレーが構成される。このことは従来の技術から知ることができる(例えば、前述の W. Hillis の "The connection machine" で説明されている。また、前述した Thinking Machine Corp. による "Connection Machine, The MIT Press, 1985, p.21, 及び前述の J. R. Nickols の "The Design of the MasPar MP-1: A Cost-Effectively-Parallel Computer" において説明されている、MasParによるMP-1などを参照)。

【0042】プロセッサ47は二つの主要なブロックである処理ユニット(PU)42と内部メモリ(INTMEM)43とに分割されている。処理ユニット42は更に3つのブロック、メイン処理ユニット(MPU)51、アドレス一致ユニット(ACU)52、及びメモリインタフェース(MEIF)53に分割されている。

【0043】マイクロコントローラ30によって伝達される一次命令および後述する方法により生成される二次命令のフォーマットには、表1に示すように、以下の図4の回路に関連するフィールドが含まれている。

【0044】

【表1】

【0049】IO:プロセッサが外部通信に使用するためのI/Oポート(61-64)を選択する。

【0050】F:現在の動作の過程で使用する条件フラッグ69を選択する。

【0051】REGSEL:現在の動作の過程で 사용되는べき内部レジスタ71を選択する(通常1つ又は2つのレジスタが一度に選択される)。

【0052】INEN:入力イネーブル信号181(図3)そのものを指定するフィールドで、これが1のときデータバス(D)32とプロセッサ内部バスL(図6)間でのデータ入力転送をイネーブルとする。

【0053】OEN:出カインーブル183(図3)そのものを指定するフィールドで、これが1のとき上記二つのバス間での出力データの転送をイネーブルとする。

【0054】図4は、図3のブロックダイアグラムにおいて示されたメイン処理ユニットMPUのブロックダイアグラムである。このメイン処理ユニットMPUの設計は、従来の技術から知ることができる(例えば、前述のC. MeadとL. Conwayによる“Introduction to VLSI Systems”, Addison Wesley, 1980, p.169参照)ので、この構成について多くは説明するつもりはない。このユニットのアーキテクチャは2つの内部データバス、Hバス(HB)101とLバス(LB)103を中心としている。内部命令バス(IIB)104は、命令バス(I)32(図1, 3)上に存在する一次命令又はそれからアドレス一致回路ACUによって後述のように二次命令が導かれたときにはその2次命令をそのアドレス一致回路から受け取る。

【0055】メインのオペランド処理は算術論理ユニット(ALU)67において行われる。処理の種類は、バスIIBから線135を介して与えられるALUに対する入力オペランドは、2つの内部データバス(HB)101と(LB)103から入力セクタ66によって与えられる。オペランドの選択は、内部命令バスIIB上に受け取った一次命令又は2次命令の中のINSELフィールドビット124によって行われる。行く先となるバスの選択は、受け取った一次又は二次の命令の中のOUTSELフィールドのビット139によって行われる。

【0056】シフター65は、内部データバスHB, LBのいずれか一方から受け取ったデータをシフトしてALUの1つの入力に与えることができる。シフトの方向(左, 右)、シフトする量(ビット単位で)、及び使われているキャリーの種類は、内部命令バスIIBにある命令のSHIFTフィールドによって指示される。

【0057】各プロセッサは4つの入出力ポート1-4(図4では、表示が混み合うのを避けるために、二つのポート61, 64だけを示している)を介してプロセッサアレー40の他のプロセッサ47と接続されている。簡単化のために、入出力ポート1-4を介したプロセッサ同士の接続は、2次元的な網目状(mesh)のネットワークを形成するようにデータリンク群114により行う。このようなネットワークでは、各プロセッサは東西南北方向の4つの一番近い隣同士とのみ接続される。したがって、データリンク群114は4つのデータリンク101, ……104からなる。しかし、これよりも複雑な接続パターンも採用してもよい。形成されるネットワークの形状は本発明の本質または応用に影響を与えることはない。

【0058】設計の簡単化のために、ALUのオペランド及び結果は1ビットとする。同様の理由で、またプロ

セッサ間の書き込みを最小に維持するために、プロセッサ間のデータリンク群114内の各データリンクはシリアルな(1ビット幅の)通信リンクを用いる。

【0059】この好ましい具体例においては、これらのデータリンク101, ……104は双方向、すなわちそのプロセッサへ、あるいは及びそのプロセッサからのいずれの方向へもデータを流すことが可能となる。これらの入出力ポート61……64の各々は、内部データバス(HB)101上に置かれたいろいろのデータソースから信号ライン113を介してデータを受け取ることができるとともに、内部データバス(LB)103に、信号ライン112を介してデータを送出することもできる。このようにして他のプロセッサからのデータは、例えばデータリンク104、入出力ポート61、ローバス(LB)103、及び入力セクタ66を含む経路を通して、ALUへ入力される。同時に、ALUの演算結果は、例えば出力セクタ68、ハイバス(HB)101、入出力ポート61、及びデータリンク101を含む経路を通してその、別のプロセッサの出力される。内部命令バスIIB(104)上にある命令は信号ライン111を介していずれかの入出力ポートたとえば、61によって受け取られ、この命令のIOフィールドによってそのI/Oポート(今の例では61)によって実行される転送の種類が規定される。

【0060】ALUによって実行される演算は、フラッグ(条件)レジスタ69の内部に従って決められる。このレジスタ69はHバス101から更新することができ、一方、現在の値を、条件テストのためにそのプロセッサのLバス103に出することも、あるいは他のプロセッサとの間でのインターロックされた条件テストのためにLバス103と入出力ポートたとえば61を介して他のプロセッサに転送することもできる。フラッグレジスタ69から現在選択されているビット、及びそのビットの転送先は、内部命令バスIIB(104)上にある命令のFフィールド149によって記述される。フラッグレジスタ69のある1ビット(Si)36はプロセッサの個々の状態ビットを表す。この状態ビットSiは、算術、論理、又はルーティングなどの演算の終了時にセットするようにしてもよい。我々の装置では、SIMDマシンではもっとも良く使用される構造すなわち、プロセッサアレー40のすべてのプロセッサの状態ビットSiのグローバルなワイヤードANDを用い、これによって得られる全体の状態ビットSでもってグローバルな同期を実現することが可能となる。マイクロコントローラ30は、例えばグローバルな状態ビットSをサンプリングする結果として、全てのプロセッサ47が自分の現在の命令を実行し終え、それぞれの状態ビットSiを立てるまでは、新しい命令をプロセッサアレー40に伝達しない。

【0061】各プロセッサ47はいくつかのスタックさ

れたレジスタ（この好ましい具体例ではそれぞれ1ビット幅）からなる内部レジスタ群（INTREG）71を有しており、これらは中間のデータを格納することができる。内部レジスタ群71はHバス101から更新することができるとともに、入力セクタ66を介してALU67へ転送するために、Lバス103に現在の値を読み出すこともできる。内部レジスタ群71から現在選択されているレジスタ及びそのレジスタに関するデータの転送の方向は、内部命令バスIIB（104）上にある命令の中のREGSELフィールドのビット138に記述されている。

#### 【0062】○アドレス一致ユニット（ACU）52

図5は、図3のブロックダイアグラムの中に示されたアドレス一致ユニットACUの内部ブロックダイアグラムである。このユニットを構成するブロックは、アドレス一致回路81、命令修正回路82、プロセッサID84、及びアドレス一致マスク（ACM）85である。

【0063】プロセッサID（識別）レジスタ84は、各プロセッサ47の物理的なアドレスを含むいくつかの比較参照値を格納するのに使用するローカルメモリである。これは図5では読み出し／書き込みができるレジスタとして示してあるが、ROM（これはプルアップ／プルダウン抵抗やヒューズのようなものを含めていろいろの実現形態がありうる）のような書き込みができないものでもよい。図5では、このレジスタとして実際には小型のRAM（ランダムアクセスメモリ）が使われる。この好ましい具体例では、プロセッサのLバス103上のデータが双方向バス168を介して書き込まれる。内部命令バスIIB（104）から導かれるアドレスライン167によって指定される位置における、プロセッサIDレジスタ84内の値PROCIDは、ライン169を介してアドレス一致回路81へ出力される。プロセッサIDレジスタ84に対して実行される動作（読み出し、書き込み）の種類は、命令バスI（31）から導かれるコントロールライン170の内容によって指示される。図26は、プロセッサIDレジスタ84の入出力信号を示す。図5に関して説明したように、これはRAMからなる。データ入力Din168は主処理ユニット51のローバスLB（103）から与えられる。命令内のビットINEN181（図6）がハイのとき、トライステートドライバ91がイネーブルされ、データはこのバス上で有効となる。このレジスタ内の適当な位置の選択は、入力Addr（167）によりなされ、この入力、命令バスI（31）のREGSELビットから与えられる。R/W信号170命令バスIから与えられる。負の書き込みパルスの立上がり時に、このレジスタの出力 Doutがラッチされ、信号PROCID（169）を次の書き込みまで与えられる。ここで我々が強調したいことは、プロセッサIDレジスタ84内の比較参照値が、プロセッサアレー40内のそれぞれのプロセッサ47に對

して固有のものだということである。場合によっては、これらの比較参照値は単にプロセッサの物理的なアドレスとすることもできるし、別の場合にはコンピュータシステムの実行開始の段階で、又はある命令を実行するのに先だって、マイクロコントローラ30によってロードされる変数とすることもできる。このように、各プロセッサ内のプロセッサIDレジスタ84の内容がそのプロセッサに固有のものであるということは、各プロセッサ40が受け取る命令の最終的な内容を決めるのに決定的に重要である。このプロセッサIDのレジスタ84はメモリマップされていなくて（すなわち各プロセッサ内の内部メモリ43（図1）をアドレスするのに使用するアドレスライン39によってアドレスされていなくて）、内部命令バス104上にある命令の中のREGSELフィールド内のビットからなるアドレス線167によって選択される。

【0064】PROCIDライン169上に与えられた比較参照値は、アドレスバス39上にあるマイクロコントローラ30から与えられた、アドレスワードの現在の値と比較される。この比較はアドレス一致回路81において行われる。アドレス一致回路81の動作は、マイクロコントローラ30から線34を介して与えられるAMODE信号34によって規定される。

【0065】この信号AMODEの詳細は後述する。この比較はビット毎に送われ、これらの比較結果に対して最終的に論理ANDが行われる。

【0066】Aバス上のアドレスワードとプロセッサIDレジスタ内の比較参照値の内、アドレス一致回路81において比較されるべきビット部分を選択するマスクがアドレス一致マスクレジスタ85によって与えられる。このレジスタ85にはLバス103（図4）から線184を介してデータが書き込まれ、そしてこの内容はライン185からアドレス一致回路81に読み出される。このレジスタ85に対する読み出し／書き込みコントロールライン171は命令バスIから導かれる。アドレス一致回路81は、上記比較の結果を、内部バス165を介して命令修正回路82に供給する。この回路は、マイクロコントローラ30（図1）から命令バスI（31）上に与えられた一次命令を、アドレスバスA（39）の選択されたビットでもって修正する。この修正の結果がローカルに有効な命令（二次命令）で、これは線163を介して内部命令バスIIB（104）（図4）に送られ、その命令修正回路82が属するプロセッサの内部ブロックに対して使用することができる。図27は、アドレス一致マスクレジスタ85の入出力信号を示す。図26と異なり、ここにはレジスタは一つしかないので、入力選択信号177のみが、命令バスI（31）のREGSELビットから与えられる。

#### 【0067】○メモリインターフェース（MEMIF）

53

図6は、図3のブロックダイアグラムに示されたメモリインターフェースユニット53の内部のブロックダイアグラムである。各プロセッサ47は、内部メモリ43（図3）と外部メモリ50（図1）のいずれも使用することができる。

【0068】RAM及びROMが内部的に使用できる空間には制限があるので、多くの応用においては外部メモリ50が必要となる。プロセッサはメモリインターフェース53内の外部メモリインターフェース54を介して、この外部メモリ50にアクセスすることができる。この外部メモリインターフェース54は外部メモリ50へ転送するデータを、Hバス101から線149を介して受け取る。このデータは双方向の外部データバス（EXTD）54Dによって外部メモリ50と交換される。外部メモリインターフェース54は、外部データバス54Dによって外部メモリ50からデータを受け取り、線155とLバス103を介してプロセッサ47の内部ブロックに転送する。データのそこへの書き込み又はそこからのデータの読み出しのために、データの転送が行われるべき外部メモリ50内の位置（アドレス）は外部アドレスライン（EXTA）54Aによって規定される。このライン54Aの内容は、アドレスバス（A）39からアドレスラッチ99、内部アドレスバス（A1）41と線153とを介して与えられる。

【0069】処理ユニット42（図3）を区切ったことの結果として、このメモリインターフェース53には外部メモリインターフェース54の他にトライステートデータバッファ91、92及び前述のアドレスラッチ99が設けてある。

【0070】トライステートバッファ91は双方向のシステムデータバス（D）32からデータを受取りローバス103に転送し、トライステートバッファ92はローバス103からデータを受け取り、システムデータバス32にデータを転送することができる。これらのバッファに対するデータ転送の指示は、命令バス31上の入力イネーブル信号INEN及び出力イネーブルOENが用いられ、これらの信号は、図5に示すようにバス31の直接の枝線としてアドレス一致回路ACU内において生成され、線181、183を介して与えられる。

【0071】システムアドレスバスA（39）上のアドレスはアドレスラッチ99によりラッチされ、内部アドレスバスA1（41）に出力され、前述のように外部メモリインターフェース54に与えられる。このラッチを指

示するアドレス更新信号NEWA46はマイクロコントローラ30によって線46を介して与えられる。タイミングに関しては、図2又は図9に示すように、システムアドレスバスA（39）がシステム命令バスI（31）より先に（たとえば、クロックサイクル0で）安定となり、その後に（たとえば、クロックサイクル5での）アドレス更新信号（たとえばクロックサイクル0で）NEWAの立ち上がりエッジでラッチ99にラッチされると仮定する。システム命令バスI（31）が変化するのはこのあと（たとえば、クロックサイクル6）からである。一次の命令からいくつかの二次命令を生々するプロセスに対して更新された内容（一致ワード）を与えるために、システム命令バスIと同時にシステムアドレスバスAは、再び変化する。しかしながら、外部メモリ50は、アドレスラッチ99からアドレスバスA1に出力されるアドレスに基づいてアクセスされ、システムアドレスバスA（39）の、この2回目の変化を見ることはない。

【0072】内部メモリ43（図3参照）は、この好ましい具体例ではデュアルポート型である。このことは、データがHバス101及びLバス103を介して2つのメモリ位置から同時に読み出し可能なことを意味する。しかしながら、メモリの衝突を回避するために書き込み用のデータはこれらのバス101又は103のいずれかからのみ与えられ、両方から与えられることは許されない。現在の内部メモリ43の動作は、内部命令バスII（104）上にある命令のIMEMフィールドによって規定される。読み出し／書き込みの目的でアクセスされるアドレスは、適当なメモリ位置を選択するために内部メモリ43の内部でデコードされる。

【0073】以下、アドレス一致回路81及び命令修正回路82の設計についての詳細について述べる。

【0074】○アドレス一致回路81

図7は、図5に示されるアドレス一致回路81の論理回路である。図7に示す好ましい具体例において、AMODE信号34はマイクロコントローラ30から与えられる2ビットの信号であり、標準的な2：4のデコーダ301において最初にデコード4ビットの信号AMODE00～AMODE11が出力される。以下の真理値表（表2）は読者の便宜を考慮してこの回路の動作を説明している。

【0075】

【表2】

表 2

AMODE	AMODE00	AMODE01	AMODE10	AMODE11
00	1	0	0	0
01	0	1	0	0
10	0	0	1	0
11	0	0	0	1

【0076】ここではアクティブ・ハイ（ハイで動作）の習慣に従う（選択された出力は論理ハイ、それ以外の出力は論理ロー）。この表から分かるように、デコーダ出力AMODE00～AMODE11はそれぞれAMODE信号34が00～11のときに論理ハイとなる。

\* 【0077】以下では、表3に示すように、AMODE信号の値によって、プロセッサアレー40に対して定義された構成モードを定義する。

【0078】

【表3】

表 3

モ 名	ド 称	AMODE	モ ー ド の 説 明
SIMD		00	これはデフォルト値であり、標準的なSIMDに対応する
命令拡張		01	この場合、アドレスバスは命令バスの拡張とみなされる
プロセッサ 選 択		10	アドレスバスは、命令の実行が可能なPEの選択に使われる
命令修正		11	各プロセッサにおいてローカルに修正した命令を生成する

【0079】ここで、これらのモードの定義についての詳しい説明を与える。

【0080】SIMDモード：これは、AMODE信号が00のときのモードである。この信号値はデフォルト値であり、このモード付標準のSIMDとしての動作モードに対応する。すなわち、アドレスバス（A）39は通常の方法で、各プロセッサ47の内部メモリ43及び外部メモリ50をアドレスするために専用的に使用される。

【0081】命令拡張モード：これは、AMODE信号が01のときのモードである。このモードでは特別なタイムスロットにおいて、アドレスバス（A）39は命令バス（I）31の拡張と見なされる。アドレスバス上の信号を一次命令の一部としてデコードする、追加の命令デコードを行わなければならない。適当なタイミングシーケンスについては後述する。

【0082】プロセッサ選択モード：これはAMODE信号が10のときである。このモードでは、アドレスバス39は、マイクロコントローラ30によって命令バス（I）31上に伝達される命令を実行すべき一部のプロセッサを選択するための情報（一致ワード）を転送する

30 のに使用される。命令は修正されない。このメカニズムは従来技術で使用されているとして前に述べたマスキングメカニズムと対比されるべきである。その従来技術では、命令を実行すべきプロセッサを選択するためには、その命令がプロセッサアレー40に転送される前に、アレー40の中の全てのプロセッサの各々内のマスキレジスタへの書き込みを必要とするが、これはかなりの時間を要する。しかし、本実施例ではこの書き込みの必要がなく、より速い動作が可能となる。本実施例では、かかるプロセッサの選択は、アドレスバス39の内容とプロセッサIDレジスタ84（図5）の出力PROCID169との一致を使う。このモードについては、後に図10についての説明において例示する。

【0083】命令修正モード：これはAMODE信号が11のときである。このモードでは、複数のプロセッサゾーンが選択され、それぞれにおいて、一次命令と異なる二次命令が生成される。アドレスワードの最下位側の10ビット、すなわち、ビット9-0は、命令修正フィールド又はプロセッサゾーンフィールドと定義されるいくつかのフィールドに分けられる（表4参照）。ここでビット0は最下位のビットである。

【0084】

【表4】

表 4

アドレスワ ード内のビ ット番号		9-8	7-5	4-3	2-0
フィールド名		第2命令修正 サゾン	第2プロセッ サゾン 選択フィールド	第1命令修正 フィールド	第1プロセッ サゾン 選択フィールド
表 記		IMOD2	PZ2	IMOD1	PZ1
例 値		11	001	01	110

【0085】すなわち、ビット2-0と7-5はそれぞれ第1、第2のプロセッサゾン選択フィールドPZ1、PZ2として用いられ、ビット4-3と9-8は、それぞれ、第1、第2命令修正フィールドIMOD1、IMOD2として用いられる。このモードでは、マイクロコントローラ30によって命令バスI(31)上へ伝達される単一の一次命令から異なるプロセッサゾンにおいて別々の命令を生成する。この結果、実質的にMIMDアーキテクチャがこのモードで得られる。このモードについては、後に図11についての説明において例示する。

【0086】図7及び表4を参照すると、各プロセッサのアドレス一致回路81においては、アドレスバス39上のプロセッサゾン選択用のフィールドPZ1、PZ2のビットと、そのプロセッサのプロセッサIDレジスタ84(図5)の出力であるPROCIDライン169の対応するビットとの一致がチェックされる。この一致チェックの結果によって、命令修正回路82が命令修正フィールドIMOD1、IMOD2のどのビットを一次命令の修正に適用すべきかを決定し、その決定されたビットでもって二次命令を生成し、ライン163を介してプロセッサの他のブロックに送付する。

【0087】アドレス一致回路81の回路レベルでの動作を理解するためには、マイクロコントローラ30によって発生される信号のタイミングを考慮しなければならない。図9は、本実施例の動作に関する信号の、より詳しいタイムチャートである。アドレスバス39は6クロックサイクルごとに更新される。クロックサイクル1から6及び13から18は、前述のSIMDモードで使用され、アドレスバス39はこの期間ではメモリアドレス情報(アドレスワード)を示している。この情報は各命令サイクル(これは、命令バス31が安定なサイクルのことであり、本実施例ではクロックサイクル7から12までの期間あるいは19から24までの期間に先だって、マイクロコントローラ30から与えられるアドレスサンプリング信号NEWA(46)を用いてメモリイン

タフェース53(図6)内のアドレスラッチ99内にラッチされる。ラッチされたアドレスは、続命令サイクルにおいて、内部アドレスバスA1(41)を介して内部メモリ43又は外部メモリ50をアクセスするのに使用される。命令サイクル(クロックサイクル7から12あるいは19から24)において、アドレスバス39は前述の命令拡張モード、プロセッサ選択モード、又は命令修正モードのために使用される。たとえば、命令修正モードでは、このバスはすでに表4に示したような、2次的な命令の生成に用いる情報(一致ワード)を示している。

【0088】図7において、一致信号COINCO110(318)は命令拡張モード(すなわちAMODE信号が01)のとき又はプロセッサ選択モード(すなわち、AMODE信号が0のとき)の動作時に特定のプロセッサで論理ハイとなる。すなわち命令拡張モード(すなわち、AMODE信号が01のとき)には、マイクロコントローラ30から受け取った一次命令が拡張されるべきプロセッサ内でのみ、信号COINCO110はハイとなる。プロセッサ選択モード(すなわち、AMODE信号が10)の時は、現在の一次命令が実行されるべきプロセッサ群に対してのみ、信号COINCO110はハイとなる。一致信号COINCO110を生成する回路は、排他的NORゲート311...314とANDゲート312...315と313とからなる。入力としてアドレスバスAの最下位側の10ビットA0-9ビット、第5図のプロセッサIDレジスタ84からPROCIDライン169を介して与えられるプロセッサIDの最下位側の対応する10ビットPID0-9及びアドレス一致マスキレジスタ85(図5)から線185を介して与えられるアドレス一致マスクの最下位側の10ビットACMR0-9を受け取る。これにより、アドレスバスAのA0-9の内の選択されたビット及びPROCIDライン169のPID0-9ビットの内の対応するビットが一致する(等しい)時のみ、論理ハイとなる。比較されるビットの選択は、ANDゲート312及び315を

介し、アドレス一致マスケジスタ85（第5図）のACMR0-9ビットの値によってなされる。アドレスA0-9とPID0-9の各ビット位置ごとの比較結果の内、ACMR0-9ビットの対応するビットが0（ロー）であるビット位置の比較結果のみが最終出力ANDゲート313によって行われる全体の比較動作に入力される。ビット毎の一致は、例えばビットA0とPID0に対しては排他的NORゲート311によって行われる。2つのビットが等しければ、排他的NORゲート311の出力はハイ（1）となる。このようなビット毎の比較結果は、対応するマスクビット（ACMR0）がゼロである場合のみ、ANDゲート312を通して、ANDゲート313による最終の比較へと伝達される。これ以外の場合には、ANDゲート312が最終のANDゲート313に対して0を伝達する。こうして、特定のプロセッサにおいてANDゲート313の出力が1となる。その後出力はANDゲート383において命令ストローブ信号INSTRENと演算され、一致信号COINCO110が生成される。この信号は、図9に示す他の一致信号COINC11-1（344）信号と同じタイミングを有するもので、命令ストローブ信号INSTRENがハイであれば間のみこの信号もハイとなりうる。以上のようにして、命令拡張モード（すなわち、AMODE信号が01のとき）には、マイクロコントローラ30から受け取った一次命令が拡張されるべきプロセッサ内でのみ、信号COINCO110はハイとなる。プロセッサ選択モード（すなわち、AMODE信号が10）の時は、現在の一次命令が実行されるべきプロセッサ群に対してのみ、信号COINCO110はハイとなる。このような一致メカニズムは、従来技術のように、それぞれの命令の供給に先だってプロセッサアレー内の各プロセッサのマスケジスタに命令の実行を行うか否かを示す情報を書き込みを行うメカニズムよりもより高速になる。命令修正モード（すなわち、AMODE信号が11）の動作においては、一致信号COINC11-1（344）及びCOINC11-2（346）がそれぞれ特定のプロセッサ群において論理ハイとなる。すなわち、ANDゲート347の出力は、プロセッサの物理的なアドレスの3つの最下位ビットが第1プロセッサゾーン選択フィールドPZ1の値110に等しいプロセッサの群（プロセッサゾーン1）に対してのみ値1を持つ。同様に、ANDゲート348の出力は、この具体例では、プロセッサの物理的なアドレスの3つの最下位ビットが第2プロセッサゾーン選択フィールドPZ2の値001に等しいプロセッサの群（第2プロセッサゾーン）に対してのみ値1を持つ。これらの一致信号COINC11-1、COINC11-2の生成を理解するためには、このモードに対応するアドレスバス39の区分を示さなければならない。表4に関して説明したように、このモードでは、アドレスワード（アドレスライン

の内容）は、命令修正フィールドIMOD1、IMOD2及びプロセッサゾーン選択フィールドPZ1、PZ2に分割される。

【0089】例をはっきりさせるために、PROCIDライン169のPIDO-9ビットはプロセッサアレー40の中の各プロセッサの物理的なアドレスを示すものと仮定する。また図7において、あらかじめ第1プロセッサゾーン選択フィールドPZ1のビットPZ10、PZ11及びPZ12及び第2プロセッサゾーン選択フィールドPZ2のビットPZ20、PZ21及びPZ22はPROCIDバス169の3つの最下位ビットPIDO-2（すなわちこの場合はプロセッサの物理的なアドレスの3つの最下位ビット）と比較されるものとする。しかしこれらのビット以外とは異なるアドレスワード中のビットを比較のために割当することももちろん可能である。

【0090】図7において、一致信号COINC11-1及びCOINC11-2の生成のための比較は、複数の排他的NORゲート341、342、345、……とANDゲート347、348によってなされる。本実施例では、ANDゲート347の出力は、プロセッサの物理的なアドレスの3つの最下位ビットが第1プロセッサゾーン選択フィールドPZ1の値110に等しいプロセッサの群（プロセッサゾーン1）に対してのみ値1を持つ。同様に、ANDゲート348の出力は、この具体例では、プロセッサの物理的なアドレスの3つの最下位ビットが第2プロセッサゾーン選択フィールドPZ2の値001に等しいプロセッサの群（第2プロセッサゾーン）に対してのみ値1を持つ。なお、第1プロセッサゾーン選択フィールドPZ1及び第2プロセッサゾーン選択フィールドPZ2に対する、アドレスワード内のビットの割り当ては、互いにオーバーラップしない又は排他的となるように、すなわち少なくとも1ビットは異なるように注意しなければならない。ANDゲート348の出力は最後のANDゲート381において命令ストローブ信号INSTRENと演算されて第9図に示すタイミングを持つ。つまり、命令ストローブ信号INSTRENがハイである期間のみハイとなる。ANDゲート347の出力は最終のANDゲート382によって命令ストローブ信号INSTRENと演算され、図9においてCOINC11-1に対して示したのと同じタイミングを持つ。

【0091】このようにして生成された一致信号COINCO110、COINC11-1、及びCOINC11-2は、デコーダ301の出力信号AMODE00（307）、AMODE01（306）、AMODE10（305）、及びAMODE11（303）とともにバス165を介して命令修正回路82へ伝達される。

【0092】○命令修正回路82

図8は、命令修正回路82の論理回路である。



25

【0093】この回路の動作は、それぞれの動作モードに分けて説明するのが最もよい。本実施例では、とくに断わらない限りマイクロコントローラ30から命令バス

(I) 29を通して到達する一次命令はビット幅が29ビットで、ビット0が最下位ビット、ビット28が最上位ビットであると仮定する。同様に、アドレスバス(A) 39のビット幅が10ビットであると仮定するが、本発明はこれらの数値以外にも適用されうることはいうまでもない。

【0094】SIMDモード(AMODE信号: 0 10 0): このモードでは命令バス31上の一次命令がそのままプロセッサ内部へ転送される。この動作は以下のようになされる。このモードでは、AMODE00信号がハイである。この結果この信号が入力されるORゲート614の出力はハイとなり、このORゲートの出力が入力されるORゲート616の出力もハイとなる。トライステートのドライバー613と612には、それぞれ命令バス31上の命令Iのビット0-1と2-28が入力されている。これらのドライバ613、612はそれぞれORゲート614、616の出力によりイネーブルされる。この結果、これらは命令バス31上にある命令を、修正しないまま、内部命令バス104(図4)に直接接続された中間バス163へ転送する。

【0095】したがって、このモードの場合には、アレー40の全てのプロセッサの各々の内部命令バス104(図4)上に現れる2次の命令は、マイクロコントローラ30から与えられた一次の命令と完全に同じになり、標準的なSIMD動作となる。

【0096】命令拡張モード(AMODE信号: 0 30 1): このモードでは、信号AMODE01(306)がハイとなる。このとき、予め選択されたプロセッサにおいては、一致信号COINCO110がハイとなっている。これらのプロセッサの選択はアドレス一致回路81の説明で示した一致メカニズム、すなわちアドレスバス39の全内容とプロセッサIDレジスタ84(第5図)の内容との間の一致メカニズムによって予め行なわれる。選択されたプロセッサは一次命令を修正した二次命令(すなわち一次命令にアドレスバスの内容を付加した命令)を受け取り、アレー40の中の非選択のプロセッサはこの一次命令を受け取る。信号AMODE01が 40 ハイとなる結果、ORゲート614の出力はハイとなり、それによってORゲート616の出力もハイとなる。こうしてトライステートドライバー613及び612がイネーブルとされる。したがって、これらのドライバは、前述のSIMDモードのときと同様に、命令バス31上にある命令を修正しないまま、内部命令バス104(図4)へ直接接続された中間バス163へと転送する。

【0097】選択されたプロセッサに対しては、オリジナルの命令に加えて、バス163及びこれに接続されて 50

26

いる内部命令バス104(図4)は、アドレスバス31から追加のビットの組(アドレスバスの全ビット情報)を受け取ることになる。このビットの組は、選択されたプロセッサにおいて、マイクロコントローラ30によって伝達されたオリジナルの命令の拡張として解釈される。より正確にいうと、拡張された命令(選択されたプロセッサに対する)のビット0から28は命令バスIからトライステートバッファ613及び612を介して与えられ、拡張命令のビット29から38はアドレスバスAの内容によって与えられる。追加のビットは、アドレスバス39からバス163へトライステートバッファ492によって転送され、このバッファ492はANDゲート495によってイネーブルとされ、このANDゲート495は入力としてAMODE01及びCOINC110信号を有し、前者は、命令拡張モード時には全てのプロセッサにおいてハイであるが後者は、選択されたプロセッサ内でハイである。内部命令バス163はオリジナルの命令バス39よりもビット幅が広くなければならず(我々の装置では39ビット対29ビットである)、そして拡張された命令の増加した能力を完全に利用するために、選択されたプロセッサにおいて追加のデコーディングを行わなければならないということが理解される。この追加のデコーディングについては後述する。

【0098】COINCO110信号がハイのときには(以下の図9に関する説明を参照)アドレスバス39は純粋なアドレス情報(アドレスワード)を含んでいないが、本発明の目的である一致プロセスに関連した情報は含んでいるのだということを指摘したい。図6の参照の際に前に説明したように、COINCO110信号がハイとなるまでには、「本当」のアドレス情報は既にラッチされており、ラッチされたアドレスバスA1上に位置している。

【0099】プロセッサ選択モード(AMODE信号: 10): このモードではAMODE10(305)信号はハイである。このモードではアレー40中の選択されたプロセッサに対するマイクロコントローラ30から伝達された一次の命令の実行はイネーブルとされ、残りのプロセッサはディスエーブルとされる。実際我々の装置では、選択されなかったプロセッサに対しては、処理を何も行わない単なるダミーの命令であるNOP(No-Operation)命令が与えられる。プロセッサの選択は、アドレス一致回路81の動作の説明において概説した一致プロセスによって行われる。以下、このモードにおける命令修正回路82の動作の詳細を説明する。

【0100】選択されたプロセッサにおいては信号COINCO110がハイとなる。(そのタイミングは図9で説明したCOINC11-1信号のタイミングと同じである)。したがって、選択されたプロセッサでは信号AMODE10及びCOINCO110はともにハイなので、これらの信号が入力されるANDゲート615の

出力もまたハイである。この出力は、ORゲート614に入力されているので、ORゲート614及び616の出力を順番にハイとする。従って選択されたプロセッサでは、トライステートドライバ613及び612はイネーブルとされる。これらのドライバは命令バスIの完全な内容を中間バス163を介して内部命令バス104(図4)上に伝える。このことは実質的に、選択されたプロセッサをイネーブルとして、マイクロコントローラ30によって伝達された一次命令を実行させることと等価である。

【0101】選択されなかったプロセッサでは信号COINCO110はハイとならないが、この結果、ANDゲート615の出力は命令サイクルの期間中(図9において、命令ストロブINTRENがローである期間)ローである。これによってORゲート614出力は、他の入力<sup>10</sup>がローであるため、ローのままであり、ORゲート616の出力もローのままである。このため、トライステートドライバ613及びB612はディセーブルされたままである。しかしながらCOINCO110がローである結果、これを入力とするインバータ630の出力はハイであり、AMODE10がハイである。このことと併せて、この信号とインバータ630の出力が入力されるANDゲート628の出力はハイとなる。この出力は、中間バス163のビット0~28の一つにそれぞれ接続された複数のトライステートバッファ611(図には一つのみ示す)をオンとする。各トライステートバッファ611の入力は、プルアップ抵抗626(これは+5ボルト(定義よりハイレベル)に接続されている)によって、ハイである。これにより、バス163のビット0から28は全て強制的にハイとなる。この好ましい装置においては、全てのビットがハイである命令は、NOP(No-Operation)命令と先験的に定義してある。したがって、以上のようにして、非選択プロセッサではNOP命令がバス163に出力されることになる。

【0102】このモードのより詳しい説明及び例については、図10の説明を参照のこと。

【0103】命令修正モード(AMODE信号:11):表1に関して説明したように本実施例で用いる命令はオペレーションコード(オペコード)部("ALU")を有する。本実施例では、このオペコード部は命令の下位6ビットにあると仮定する。修正命令生成モードでは、命令修正回路82はマイクロコントローラ30から転送された命令の下位2ビット(すなわち、オペコード部の下位2ビット)を修正することにより、新たなオペコードを有し、他の部分は元の命令と同じである、新たな命令を生成する。さらに、新たに生成される命令は、第1、第2プロセッサゾーンのそれぞれにより異なる。このモードでは、AMODE11(303)信号はハイである。第8図において、I0(441)及びI1(442)は、命令バス31の命令の2つの最下位ビッ

ト0、1ある。同時に、IMOD1-0及びIMOD1-1は第1命令修正フィールドIMOD1に含まれるビットであり、IMOD2-0及びIMOD2-1は第2命令修正フィールドIMOD2に含まれるビットである。

【0104】図8において、オリジナルの命令のビットI0及びI1は、それぞれORゲート461、462によって、第2命令修正フィールドIMOD1のビットIMOD1-0、IMOD1-1との間でOR演算がなされ、さらにORゲート463、464によって第2命令修正フィールドIMOD2のビットIMOD2-0、IMODE2-1との間でOR演算がされる。これらのORゲートの出力はトライステートドライバ453、454、451、452がアクティブのときに、中間バス163に出力される。ところで第1プロセッサゾーン内のプロセッサでは、前述したように一致信号COINC11-1がハイであるので、修正命令生成モードでは、ANDゲート402の入力COINC11-1とAMODE11がともにハイであるため、このANDゲート402の出力はハイとなり、ドライバ453、454とともにアクティブとする。この結果ORゲート463、464の出力がバス163に出力される。同様に第2プロセッサゾーン内のプロセッサでは、ANDゲート401がオンとなり、ドライバ451、452がともにアクティブとなり、オアゲート461、462の出力が中間バス163に出力される。各PEは第1プロセッサゾーンもしくは第2プロセッサゾーンのどちらか一方に戻し、その情報をプロセッサIDレジスタ84内に持つ。プロセッサIDレジスタ内のゾーン情報はPROCID信号によりアドレス一致回路81に送られアドレスバス内のゾーン指定PZ1およびPZ2と比較され一致した場合COINC11-1またはCOINC11-2が1となる。このように、ドライバ451と452の組と453と454の組の内、一方の組のドライバのみが一時にアクティブとなり、従ってバス163上でこれらの組の出力が衝突することはない。こうしてオアゲート461、462の組又は463、464の組の出力が二次命令の下位2ビットとしてバス163に出力される。

【0105】ここで、命令修正ビットによるオリジナルの命令ビットの修正(「変調」という語を使用することもできる)のためにORゲート461~464を使用することは、このプロセスに使用できる多くの選択枝の中の一つに過ぎないということを述べておきたいと思う。

【0106】この具体例では、命令バスI31の最下位の2ビットだけが命令修正回路82によって演算され、残りの部分は修正されない。しかしながらこれは既に、自分が含まれるプロセッサゾーンに基づいてそのプロセッサに対して使用される新しい命令の組をいくつか持っていることと等価である。プロセッサアレーの区分けは、それぞれ異なる最終的な命令を受け取る異なるゾー

29

ンにあるプロセッサ47に関して行われる（マイクロコントローラ30から命令バス39を通して伝達されるオリジナルの命令に対して、最終的な命令によって、プロセッサの内部命令バス（IIB）51上に位置する最終的な命令を理解する）。ゾーンは書き込み可能のPROCIDによって規定されるので、アレーに対する非常に多変形の変形が可能であり、一つの命令から次の命令へと大きく変化する。

【0107】図9及びコントロール信号のタイミングシーケンスについて前のセクションで与えた説明を参照すると、一致信号（例えば、COINC11-1）が安定である時は、内部命令バス104（図4）はクロックサイクル7から12及び19から24（ゲートによる小さい遅延は無視する）の期間中は安定であることが分かる。この一致信号は図8に示すトライステートドライバ（453や454など）をイネーブルにする（ハイの時に）。

【0108】このモードの更に詳しい説明や例については、図11の説明を参照のこと。

【0109】図9に示すように、アドレスバス39は6クロックサイクルごとに更新される。クロックサイクル1から6及び13から18においては、アドレスバス39はアドレス情報（アドレスワード）を示している。この情報は、各命令サイクル（命令ライン31が安定である、クロックサイクル7から12及び19から24）に先だってアドレス更新信号NEWA信号を用いてラッチ99（図6）にラッチされる。ラッチされたアドレスは、バスA1（図6）を介して使用可能である命令サイクルにおいてはアドレスバス（A）39の内容は異なる。すなわち、ラッチされていないアドレスバス39の内容とPROCIDラインとの間の一致プロセスに基づいてプロセッサアレーの区分け（2次的な命令の生成及び実行に関数する）のために使用する情報（一致ワード）を示している。

【0110】図10は、提案された回路の可能な動作モードの一つであるプロセッサ選択モードを理解するのに役立つ図式的に示したものである。このモードではアドレスバス（A）39は、イネーブルとされてマイクロコントローラ30からプロセッサへ伝達された命令を実行するプロセッサを選択するために使用される。命令は修正されない。図に示した例で8×8個のプロセッサのみを持っているプロセッサアレーとしたのは、便宜を考慮したためである。この図は事実上プロセッサ選択モードでは、アレー40内の各プロセッサによって実行される命令471のマップ401である。ここでマイクロコントローラ30から伝達されるオリジナルの命令が、EX-OR（排他的論理和）命令であると仮定する。図を見ると分かるように、アレー40の上から6列はオリジナルの命令EX-OR（472）を実行するようイネーブルとされており、一方最後の2列は、ダミー命令で

30

実式的に効果を生じないNOP（No-Operation）命令を実行する。このことは図8を参照すると説明できる。アレーの中の上の6列のプロセッサに対しては、一致信号COINC0110は論理1であり、下の2列に対しては論理0である。このことは前に説明したように、各プロセッサのプロセッサIDレジスタ84及びアドレス一致マスクレジスタ85の中に適当な値をセットすることによって得られる。信号COINC0110がハイである結果、そしてモード信号AMODE10もまたハイであるために、上の6列のプロセッサ（これが選択されたプロセッサである）に対して、ANDゲート615（図8）の出力はハイとなり、これによってORゲート614及び616の出力もまたハイとなる。この結果、トライステートバッファ613及び612はオンとなり、マイクロコントローラの命令バスI上の次命令の全ビットをバス163に伝達し、もってこの命令を内部命令バス（IIB）104（図4）に乗せる。このように2次的な（導出）命令（すなわち内部命令バス（IIB）104上にある命令）は、マイクロコントローラ30によって伝達されるオリジナルの命令の等しいコピー、すなわちEX-OR命令である。

【0111】アレーの中の下2列のプロセッサ（選択されていないプロセッサ）については、ANDゲート615の出力は、COINC0110がローであるのでローとなり、これによりORゲート614及び616もまたローとなる。この結果、トライステートバッファ613及び612はオフとなり、オリジナルの命令が内部命令バス163へ転送されるのを遮断している。しかしながらインパータ630の出力は、アレーの最後の2列のプロセッサに対してはハイとなり、これとモード信号AMODE10の値がハイであることから、ANDゲート628の出力をハイとし、これは複数のトライステートバッファ611（図では一つのみ示す）を順次オンとする。これらトライステートバッファ611の入力は、プルアップ抵抗626（定義より+5ボルトの電圧に接続されている）によってハイである。これによりバス163の全てのビット0から28の値はハイとなり、これは順次内部命令バスIIBに転送される。全てのビットがハイである内部命令は、ここでの好ましい装置の内部の命令の組では、NOP（No-Operation）命令と定義されている。これらより図10のマップが導かれ、下2列のプロセッサはダミーのNOP動作を実行し、事実上の効果は生じない。

【0112】第11図は、命令修正モードの理解に役立つ図式的な表示である。このモードではマイクロコントローラ30によって命令バス31上に伝達された単一の命令から開始して異なるプロセッサにおいて別々の命令を生成することを可能とする。第11図の例において、プロセッサアレーは便宜のために8×8個のみ図示してある。この図は事実上、モード11の時のアレー内の各

プロセッサによって実行される命令のマップ401である。マイクロコントローラ30によって伝達されるオリジナルの命令はEX-OR（排他的論理和）命令であると仮定する。図よりプロセッサアレー40の中のプロセッサは2つのゾーンに区分けされ、一方（図11の外側ゾーン）は2次的（導出）命令である+（加算）命令475を実行し、他方（図11の内側ゾーン）は2次的（導出）命令である（減算）命令477を実行する。これは図9へ戻って参照すれば説明することができる。図8において、外側ゾーンのプロセッサにおいて一致信号COINC11-1はハイで、内側ゾーンのプロセッサに対してはローであると仮定する。同時に、一致信号COINC11-2は外側ゾーンのプロセッサにおいてはロー、内側ゾーンのプロセッサに対してはハイである。これは前に説明したように、図7の回路において適当な一致をトリガーするために、各プロセッサ内のプロセッサIDレジスタ84及びアドレス一致マスクレジスタ85に適当な値をセットすることによって、得られる。アドレス一致回路81を参照して説明した我々の例に関しては、外側のゾーンにあるプロセッサはプロセッサゾーン1（PZ1）に属し、内側のゾーンにあるプロセッサはプロセッサゾーン2（PZ2）に属する。

【0113】図8より分かるように、PZ1にあるプロセッサは、内部命令の最初の2つのビットに対して一組のトライステートドライバー453及び454の出力を使用する（COINC11-1のハイレベルによりこれらのドライバーをイネーブルにする）。同時に、PZ2\*

10

20

\*にあるプロセッサは内部命令の最初の2ビットに対して他の組のトライステートドライバー451及び452の出力を使用する（COINC11-2のハイレベルによりこれらのドライバーをイネーブルにする）。各組のトライステートバッファ451、452又は453、454の入力側に位置するORゲート461、462又は463、464は、適当な命令修正ビットIMODによってオリジナルの命令ビットIOおよびI1の変調を生成する。各組のトライステートドライバーによって、中間バス163を介して内部命令バス104上に位置された命令は、このように命令修正フィールド（プロセッサゾーン1におけるプロセッサに対してはIMOD1、プロセッサゾーン2におけるプロセッサに対してはIMOD2）に対応して修正された最初の2つのビットを有している。

【0114】図11に示す場合には、オリジナルの命令（EX-OR命令）は、算術論理ユニット（ALU）67（図4）によってデコードされる2つの2次的な（導出）命令に変換される。これらはそれぞれ+（換算）命令及び-（減算）命令である。ALU67の構成に基づいて、2次的な命令をデコードする種々の可能性が考えられる。

【0115】表5は、マイクロコントローラによって伝達される最初の命令といくつかの2次的な命令との関連を示している。

【0116】

【表5】

表5 (AMQDE11=1)

一 次 命 令					二 次 命 令				
10	11	12	13	129	10	11	12	13	129
0	0	1	0	1 (EX-OR)	0	1	1	0	1 (+)
					1	1	1	0	1 (-)
					1	0	1	0	1 (AND)

【0117】ここで示されているのは、図11を参照して上で議論したものと同一のものであり、29ビット長の最初の（primary）及び2次的な命令が仮定されている（IOからI28）。この場合最初の命令はEX-OR命令であり、この命令から導出可能な3つの命令は+（加算）、-（減算）、及び論理ANDである。これら4つの命令は最下位の2ビットIOおよびI1を例外として、それ以外のビット構成は同一である。これら2つのビットIO及びI1は、プロセッサゾーンのビットとPROCIDライン（上記図11の説明参照）との一致に基づいて、命令回路によって対応する命令修正部分（表4）と置き換えられる。この好ましい具体例では、各最初の命令に対して一度に3つの置き換え（導出命

40

50

令）ができるが、アドレスワードの構造（表4）においてたった2つのプロセッサゾーンしかないので、2つの2次的な命令だけを同時にプロセッサアレー中で使用することができる。

【0118】プロセッサアレーの命令の全ての組は、それぞれが4つの命令を含む、命令のサブセットに区分けされることが分かる。サブセットのなかの4つの命令のそれぞれは、最初の命令としてマイクロコントローラによって伝達され、そしてサブセットの中の残りの3つの命令の中から2つを置き換えることができる。

【0119】それぞれの命令が他の命令によって置き換えがなされるためには、命令修正ビットの部分は2ビットよりも幅が広くなければならない。例えば、64個の

最初の位置を含んでいる命令セットにおいて完全な置き換えを行うためには、64種類のデコードを行うために命令修正ビットは6ビットでなければならない。この場合に命令修正回路に対して便利な構成は、ROM（リードオンリメモリ）とすることである。このROMに対する入力（アドレス）は現在のプロセッサゾーンに対応する命令修正ビットとなり、ROMからの出力は2次的な（導出）命令となる。このROMはアレーの中のプロセッサのそれぞれについて格納しなければならないので、より進んだ集積回路技術が使われるようになって初めて現実のものとなる。

【0120】ここでの具体例では2次的な命令はALU部分のみが最初の命令と異なっているが、命令の他の部分に対する修正（INSEL, OUTSEL, SHIFT, IO, REGSEL, INEN, 及びOEN）も同様に行うことができる。

#### 【0121】○変形例

当業者にとっては明らかなように、本発明の思想及び範囲の中において上で説明した具体例に対する多くの変形が可能である。

【0122】1次的な命令から2次的な命令が生成される方法に関して「生成」という話を使う場合には常に、特別な場合として2次的な命令がオリジナルの（1次的な）命令と等しい場合を含む。

【0123】複数の2次的な命令のローカルな生成及び実行を行うことを目的として、プロセッサアレーの再配置のために必要とされる余分な情報を伝達する搬送手段としてアドレスバスを使用することは、有利なことであるが（既にシステムの中で配線されているバスを使うのだから）、しかし本発明はこれに限らない。たとえば、かわりにマイクロコントローラによって駆動される別のコントロールバスを使用してもよい。この方法の不利な点は、この追加したコントロールバスをアレーの中の全てのプロセッサに対してルーチングする必要があるということである。しかしこの場合、アドレスバスはこれ以上時分割多重（time-multiplexed）しないので、マイクロコントローラによって生成される信号のタイミングシーケンスはより簡単となる。またアドレスバスをラッチする必要もなくなるという利点を有する。

【0124】図2及び図4に示すプロセッサ47のダイアグラムもまた、単に一例を示すに過ぎない。ALU内のビット数及び通信ポートの数もまた選択できる。

【0125】本発明は、プロセッサアレー40の中のプロセッサ全てを接続するのに使用する相互接続ネットワークとは独立して実施することができる。われわれの説明では、網目状のネットワークが用いられ、最も近い隣同士を接続したが、これも変形することができる。また、プロセッサどうしの間でデータの交換に使用したルーチングも変形できる。これを回路スイッチ式、又はパケットスイッチ式とすることもでき、絶対的な又は相対

的なアドレッシングに基づくこともできる。

【0126】好ましい具体例において提案された相互接続が、ワイヤを基礎とした相互接続技術に使用されているが、コントロール信号又はデータのワイヤレスでの配置（例えば光学的な手段によって）も本発明の範囲である。

【0127】ホストコンピュータ20及びマイクロコントローラ30は種々の構成が可能であり、これらを結合して1つのユニットにもできる。

【0128】プロセッサアレー40は長方形として示されているが、パッケージング技術及び使用される相互接続ネットワークに基づいて、別の配置を使用してもよい。

【0129】プロセッサIDレジスタ84の内容は必ずしもアレーの中のプロセッサの現在の物理的なアドレスと関係がある必要はない。プロセッサアレーをプロセッサゾーンに有効に分割するのに有用と思われる任意のマスクパターンを使用することができる。これに関連してプロセッサゾーンとは、単純に、同じ2次的な（派生した）命令を同時に実行するプロセッサのグループとして定義される。

【0130】プロセッサIDレジスタ84は読み出し－書き込み機能を持つことができるので（RAM又はレジスタが最も直接的である）、プロセッサアレーの分割は、アレー中にマップされたアルゴリズムの必要性に依存して、一つの命令ごとにダイナミックに変更することもできる。

【0131】アドレスバス及びプロセッサIDレジスタ84のビットの内、一致を検出すべきビットの割り当てを変更することもできる。これはまた、割り当てがダイナミックに変化する、再プログラム可能な装置とすることもできる。

【0132】我々が提案した本発明の具体例においては、全てのアドレスバスが全てのプロセッサに接続されていた。アドレスバスがプロセッサに接続される前にデコードされる場合には、共通のチップセレクトラインがアレーの同じ空間領域にある全てのプロセッサに接続されてよい（すなわち同じ列、同じ行、又は同じプリント基板内の全てのプロセッサに対する共通のチップセレクト信号）。この場合でも我々の発明は適用され得るが、プロセッサゾーン内のアレーの区分けを規定する際に得られる空間的な分割は、それぞれのチップセレクトによってコントロールされる領域（列、行、又は基板）に帰着される。この場合、個々のアドレスビットの場合とは反対に、一致検出プロセスはデコードされたアドレス信号（チップセレクト）によって指示される。

【0133】好ましい具体例は、我々の発明が適用される並列コンピュータに対するSIMDモデルを仮定している。しかしながら、MIMDマシンへの拡張も可能である。このようなマシンでは、マイクロコンピュータは

いくつかの命令の流れをプロセッサアレーへ与える。このような環境において本発明を適用することによって、元々同じ命令を受け取るプロセッサがローカルには別々の2次的な命令を生成することができる可能性へと通じる。このようにして、計算の粒度 (granularity) (すなわち全て同じ命令を実行するプロセッサのドメインの寸法) を減少させることができ、これにより並列アレーにおいて、より複雑な並列アルゴリズムを搭載することができる。

【0134】さらに、他の変形例として本発明はMIMD構成の並列プロセッサにも適用可能である。図13において例示したように、MIMDプロセッサアレー399はいくつかのプロセッサの基本的なグループに分割され、各グループは共通の命令を受け取る。例えば、図13のプロセッサアレー399では、基本的な一つのプロセッサグループ821に属するプロセッサは全て同じ命令(11)813を受取り、一方基本的な他のプロセッサグループ823の中のプロセッサは全て同じ命令(16)817を受け取る。同時に、プロセッサグループ811に対応するアドレスバスは(A1)811、プロセッサグループ823に対応するアドレスバスは(A6)816である。全てプロセッサは、先に実施例で述べたように、アドレス一致回路及び命令修正回路を含んでいる。それぞれの基本的なプロセッサグループは自分のアドレスモードラインを持ち、プロセッサグループ821に対してはAMODE18ライン14、プロセッサグループ823に対してはAMODE68ライン18を有している。

【0135】図14に示したように、我々の発明を別々に基本プロセッサグループに適用した結果、2次的な命令は対応する最初の命令流から各プロセッサにおいてローカルに生成され、命令修正情報は対応するアドレスバスに含まれる。従って各プロセッサグループはより小さい2次的なプロセッサグループに分割され、各2次的なプロセッサグループは同じ派生命令を実行する。このように基本プロセッサグループ821内のプロセッサが2つの2次的なプロセッサグループ831(派生命令NOPを実行)及び832(派生命令EX-ORを実行)に分割されるが、これらは同じ一次的命令を受け取る。同時に、基本プロセッサグループ823も2つの2次的なプロセッサグループ834(導出命令+、算術和を実行)及び835(導出命令EX-OR、排他的論理和)に分割されるが、これらは同じ一次的命令を受け取る。

【0136】結果として、計算の粒度は、同じ最初の命令を実行する3×3個のプロセッサからより小さいサイズへ縮小する。したがってプロセッサアレー内のより複雑なマッピング(空間的分解)を必要とするアルゴリズムでも、アレーにおいてうまくマッピングされる(マッピングとは、元々のアルゴリズムからプロセッサの空間的物理的な配置に対して命令を配置させることであ

る)。

#### 【0137】○他の実施例

上記の実施例(以下、第1の実施例と呼ぶ)では、①命令の修飾能力がアドレスバス幅で制限される。②命令の修飾情報をマイクロコントローラ30で生成するため、従来例に対して、マイクロコントローラ30を大幅修正する必要がある。あるいは、応用ごとにマイクロコントローラ30を変える必要がある。③マイクロ命令31とアドレスバス39のデコード回路を全てのPE内部に持つため、プロセッサアレー40の面積が必要以上に大きくなる。等の問題点がある。そこで、第2の実施例を以下に示す。

【0138】図15は、第1の実施例と第2の実施例の違いを示すためのシステム構成の概略図である。第1の実施例と第2の実施例の違いを示すために不要な信号線は省略している。SIMDの拡張を行う場合、図15(a)では、マイクロ命令31をアドレスバス39を利用した情報で修飾して各PEの演算機能を決定する。修飾の有無はAMODE34で指定する。命令修飾情報はマイクロコントローラ30で生成する。演算機能を決定するための回路は全てのPE内に持つ。一方、図15(b)では、SEL1031の信号値に従い、予めチップ内に準備された拡張機能記憶装置1200からの命令のみにより各PEの演算機能が決定する。拡張機能記憶装置1200には、既にデコードした後の信号を格納可能である。マイクロコントローラ30には命令修飾情報を生成するための修正を行う必要はない。各PE内で命令のデコードは不要である。従って、第1の実施例が持つ上記①～③の問題点を対策可能である。以下、第2の実施例を詳細に示す。

【0139】図16は、各PEの列ごとに一つの拡張機能記憶装置を対応させたプロセッサアレー40の一例を示している。図16において、1511～1514、1521～1524、1531～1534、1541～1544はプロセッサアレー40を構成するPEである。1301は拡張機能指定信号であり、全PEが同一演算機能を実行するか、あるいは各列ごとに異なる演算機能を実行するかを指定する。1302は通常のSIMDアーキテクチャ時のALUの演算機能指定信号であり、マイクロ命令31で与える例を示している。1100は命令決定回路であり、拡張機能記憶装置1200を含んでいる。命令決定回路1100は、信号1301、1302および拡張機能記憶装置1200からの情報をもとにALU制御信号1410～1440およびNOP(No Operation)指定信号1416、1426、1436、1446を出力する。

【0140】図17に命令決定回路1100の一例を示す。図17において、命令決定回路1100は、デコード回路DEC1120、拡張機能記憶装置1200、セレクト2410～2440により構成されている。拡張



機能記憶装置1200はRAM、EPROM等の書替え可能なメモリまたはレジスタで構成されている。図17の例では、拡張機能記憶装置1200は1210～1240の4個のサブグループに分かれている。各サブグループは、それぞれ異なるPEグループ(1511～1514、1521～1524、1531～1534、1541～1544)に対応している。デコーダ回路DEC1120は、信号1302を、通常のSIMDアーキテクチャ時のALUの演算機能指定信号とする場合のデコーダ回路である。図17の例では、信号1302は拡張機能記憶装置1200のアドレス信号も兼ねている。セクタ2410～2440は拡張機能指定信号1301に従い、予め拡張機能記憶装置1200に格納された命令(2210～2240)とマイクロ命令で指定された命令(デコーダ回路DEC1120の出力信号2120)のどちらをALU制御信号1410～1440とするかを選択する。2210は拡張機能記憶装置1210に格納された5ビットの信号値と、同じく拡張機能記憶装置1210に格納された信号値1216との論理積をとった信号である。2220は拡張機能記憶装置1220に格納された5ビットの信号値と、同じく拡張機能記憶装置1220に格納された信号値1226との論理積をとった信号である。2230は拡張機能記憶装置1230に格納された5ビットの信号値と、同じく拡張機能記憶装置1230に格納された信号値1236との論理積をとった信号である。2240は拡張機能記憶装置1240に格納された5ビットの信号値と、同じく拡張機能記憶装置1240に格納された信号値1246との論理積をとった信号である。1416～1446はNOP指定信号である。

【0141】以下、命令決定回路1100の機能を説明する。命令決定回路1100は、マイクロ命令31から入力した拡張機能指定信号1301の信号値が0の場合、マイクロ命令31から入力した演算機能指定信号1302をDEC1120でデコードした信号2120の信号値を、それぞれ出力信号1410～1440に伝える。出力信号1416、1426、1436、1446の信号値は0とする。命令決定回路1100は、マイクロ命令31から入力した拡張機能指定信号1301の信号値が1の場合、マイクロ命令31から入力した信号1302を拡張機能記憶装置1200のアドレス情報として、拡張機能記憶装置1210に格納された5ビットの信号値とNOP指定信号1216との論理積を、出力信号1410に伝える。同様に、1220～1240に格納された5ビットの信号値とNOP指定信号1226、1236、1246との論理積を、それぞれ出力信号1420、1430、1440に伝える。出力信号1416、1426、1436、1446の信号値は、それぞれNOP指定信号1216、1226、1236、1246の信号値が0の場合のみ、1とする。

【0142】図18にセクタ2410の詳細回路図を示す。セクタ2420～2440も同一構成である。セクタ2410は拡張機能指定信号1301の信号値が0の場合、信号2121～2125の信号値を、それぞれ出力信号1411～1415に伝える。出力信号1416の信号値は0となる。拡張機能指定信号1301の信号値が1の場合、信号2211～2215の信号値を、それぞれ出力信号1411～1415に伝える。出力信号1416の信号値はNOP指定信号1216の信号値が1のとき0、0のとき1となる。ここで、信号2211～2215は、それぞれ情報記憶手段1210に格納された5ビットの信号値とNOP指定信号1216との論理積をとった信号である。

【0143】図19に算術論理演算ユニット(ALU)1610の一例を示す。図19において、1611と1612はALU1610の入力信号であり、1613は出力信号である。1411～1415はALU1610の制御信号である。5本の制御信号のうち、1411～1413の信号値により、AND、OR、EOR、ADDの4つの演算機能の中から一つを選択する。制御信号1414の信号値が1のとき、スイッチ1063がオン状態となり、入力信号1612の信号値(S)が演算処理に使用される。制御信号1414の信号値が0のとき、スイッチ1064がオン状態となり、入力信号1612の信号値を反転した値(¬S)が演算処理に使用される。制御信号1415の信号値が1のとき、スイッチ1061がオン状態となり、入力信号1611の信号値(R)が演算処理に使用される。制御信号1415の信号値が0のとき、スイッチ1062がオン状態となり、入力信号1611の信号値を反転した値(¬R)が演算処理に使用される。ただし、スイッチ1061～1064がオンとなるのは、クロック信号PH1が1のときだけである。制御信号1411～1415の信号値と演算処理機能(ALU機能)およびマイクロ命令での対応するコードの関係を表6に示す。

【0144】表6は、8種類の演算処理機能(ALU機能)に対して、各演算処理機能を実現するための制御信号1411～1415の信号値(5ビット)およびマイクロ命令での対応するコード(3ビット)を示している。

【0145】

【表6】



39  
表 6

マイクロ コード	ALU機能	ALU制御信号
1		1 1 1 1 1
3		4 4 4 4 4
0		1 1 1 1 1
2		5 4 3 2 1
0 0 0	R + S	1 1 1 1 0
0 0 1	R - S	1 0 1 1 0
0 1 0	$R \oplus S$	1 1 0 1 0
0 1 1	$R \odot S$	1 0 0 1 0
1 0 0	S - R	0 1 1 1 0
1 0 1	$R \wedge S$	1 1 0 1 1
1 1 0	$R \vee S$	1 1 0 0 0
1 1 1	$R \wedge S$	1 1 0 1 1

【0146】従って、制御信号1411~1415の信号値(5ビット)を、各PEごとに予め図17に示す拡張機能記憶装置1200に格納しておくことにより、拡張機能指定信号1301の信号値が1の場合、各PEグループごとに異なる演算処理機能を実行可能である。

【0147】図20は、PEグループの数が4つの場合を例にとり、拡張機能記憶装置1210~1240に格納したコードと各PEグループの演算処理機能の関係を示している。図20において、40はプロセッサアレイ、1510~1540はPEグループを示している。拡張機能記憶装置1210~1240の最上位ビットはNOP指定ビットであり、0のときNOP指定である。拡張機能記憶装置1210~1240の下位5ビットは表6に示した制御信号1411~1415の信号値(5ビット)である。すなわち、拡張機能記憶装置1210~1240のアドレス0を指定すると、PEグループ1510と1520は加算を実行し、PEグループ1530と1540は減算を実行する。アドレス1を指定すると、PEグループ1510、1520と1540は加算を実行し、PEグループ1530はNOPとなる。アドレス2を指定すると、PEグループ1510は論理積を実行し、PEグループ1520、1530と1540は論理和を実行する。アドレス3を指定すると、PEグループ1510は加算を実行し、PEグループ1520は減算を実行し、PEグループ1530は排他的論理和を実行し、PEグループ1540は論理和を実行する。図20の例では、このような組合せを全部で8種類拡張機能記憶装置1210~1240に格納できる。

【0148】表7に、図17に示したデコーダ1120の真理値表を示す。

【0149】

【表7】

40  
表 7

マイクロ コード	ALU制御信号 コード
1	2 2 2 2 2
3	1 1 1 1 1
0	2 2 2 2 2
2	5 4 3 2 1
0 0 0	1 1 1 1 0
0 0 1	1 0 1 1 0
0 1 0	1 1 0 1 0
0 1 1	1 0 0 1 0
1 0 0	0 1 1 1 0
1 0 1	0 1 0 1 1
1 1 0	1 1 0 0 0
1 1 1	1 1 0 1 1

10

20

30

40

50

【0150】入力信号1302で指定したコードに従い、表6に示したALU制御信号1411~1415の8種類のパターンのうちの 하나가選択される。すなわち、表6に示した8種類の演算処理機能(ALU機能)のうちの 하나가選択される。

【0151】図21は、図17に示した命令決定回路1100を用いた他の並列プロセッサの一例である。図21において、40は並列プロセッサを実現するLSIであり、1510~1540は並列プロセッサを構成するPEである。LSI40は、マイクロ命令31で指定した拡張機能指定信号1301および演算機能指定信号1302を命令決定回路1100の入力として、PE1の演算機能の制御信号1411~1415およびNOP指定信号1416を生成する。同様に、PE2の演算機能の制御信号1421~1425およびNOP制御信号1426を生成する。PE3の演算機能の制御信号1431~1435およびNOP制御信号1436を生成する。PE4の演算機能の制御信号1441~1445およびNOP制御信号1446を生成する。

【0152】図22はPE1の一構成例である。PE2~PE4も同一構成である。図22において、1610は算術論理演算装置(ALU)、1611(R)と1612(S)はALU1610の入力信号、1613は出力信号である。1620はシフタ(SFT)、1630は汎用レジスタ(REG)、1640は演算の途中結果を格納する作業用レジスタ(REG(W))、1650は他PEからの入力データを一時的に格納する入力バッファ(IB)、1660は他PEへの出力データを一時的に格納する出力バッファ(OB)である。1651は他PEからの入力データを伝えるための配線、1661は他PEへの出力データを伝えるための配線である。1670および1680は、REG1630、REG

(W) 1640またはIB1650からALU1610への入力データを伝えるためのバス、1690は、ALU1610の演算結果をREG(W) 1640またはOB1660へ伝えるためのバスである。32はPE1とホスト計算機とのデータの授受を行なうためのデータバスである。PE1を制御するための制御信号は省略している。

【0153】図23は、図22に示すPEを制御するための、マイクロ命令のフォーマットの一例である。図23において、フィールド1はバス1670への入力元およびバス1670からの出力先を示す。フィールド2はバス1680への入力元およびバス1680からの出力先を示す。フィールド3はバス1690への入力元およびバス1690からの出力先を示す。フィールド4はバス32への入力元およびバス32からの出力先を示す。フィールド5はALUの演算機能を制御する。ALUの演算機能制御部のうち、入力制御部はALUの入力マルチプレクサ(MUX)の制御を行う。機能指定部はALU単独の算術演算および論理演算を指定する。出力指定部はシフタ(SFT)に関する制御を行う。マイクロコードと制御機能の対応の一例をそれぞれ表8、表9および表10に示す。

【0154】

【表8】

表 8

マイクロ コード	ALUソース オペランド	
	R	S
I <sub>2</sub> I <sub>1</sub> I <sub>0</sub>		
0 0 0	バス A	バス A
0 0 1	バス A	バス B
0 1 0	バス A	0 固定
0 1 1	0 固定	バス A
1 0 0	バス D	バス A
1 0 1	バス D	バス B
1 1 0	バス D	0 固定
1 1 1	0 固定	バス B

【0155】

【表9】

表 9

マイクロ コード	ALU機能
	I <sub>5</sub> I <sub>4</sub> I <sub>3</sub>
0 0 0	R + S
0 0 1	R - S
0 1 0	R ⊕ S
0 1 1	R ⊙ S
1 0 0	S - R
1 0 1	$\overline{R} \wedge S$
1 1 0	R ∨ S
1 1 1	R ∧ S

【0156】

【表10】

表 10

マイクロ コード例	S F T 入力	マイクロ コード例	S F T 機能	マイクロ コード例	S F T 機能	マイクロ コード例	作業用 R E G
I <sub>10</sub>		I <sub>9</sub> I <sub>8</sub>		I <sub>7</sub>		I <sub>6</sub>	入力
0	ALU	0 0	シフト無し	0	バスCへ 出力	0	ALU
		0 1	右シフト				
1	バスB	1 0	左シフト	1	バスCへ 出力せず	1	バスC
		1 1	シフト未通過				

【0157】前記演算機能指定信号1302は、表8のI<sub>0</sub>～I<sub>2</sub>に対応する。拡張機能指定信号1301は、図23に示すマイクロ命令のフォーマットに追加する必要がある。

【0158】図24は、マイクロ命令31で指定した、ALUの機能指定コード1032および拡張機能指定コード1031と並列プロセッサ40を構成するPEグループ1510～1540の演算機能との対応を示している。拡張機能指定コード1031が0のとき、PEグループ1510～1540は全て同一の演算を実行する。実行する演算は、表6に示したマイクロコード（ALUの機能指定コード1032）とALU機能の対応表に従う。拡張機能指定コード1031が1のとき、ALUの機能指定コード1032は図20に示した拡張機能記憶装置1210～1240のアドレス指定として使われる。すなわち、予め拡張機能記憶装置1210～1240に格納されたALU制御信号コードに対応するALU機能が、それぞれのPEグループで実行される。ALU制御信号コードとALU機能の対応は表6に従う。

【0159】以上の実施例では、各PEグループごとに拡張機能記憶装置を対応させる例を示した。各PEごとに拡張機能記憶装置を対応させることも可能である。

【0160】以上の実施例では、拡張機能記憶装置をLSI内の1ヶ所にまとめて持つ例を示した。各PEごとに前記拡張機能記憶装置を持つことも可能である。

【0161】図25は、各PE内のレジスタの一部を情報記憶手段として用いた他の実施例を示している。図25において、40はプロセッサアレイ、1510～1540は並列プロセッサを構成するPEである。1301は拡張機能指定信号である。1302はALUの演算機能指定信号であり、マイクロ命令31で与える例を示している。1110はデコード回路であり、拡張機能記憶装置のアドレス選択信号1111を出力する。1120は図17に示したデコード回路1120であり、ALU制御信号2120を出力する。32はホスト計算機とPEとの間のデータの授受を行うためのデータバスである。図25では、PE1510を例にとり、拡張機能記

憶装置1210に関連する情報を図示している。PE1520～1540も同一構成である。PE1510において、1610は図19に示した算術論理演算ユニット（ALU）であり、1210は拡張機能記憶装置をレジスタ群の一部を用いて構成している。拡張機能記憶装置1210内へのALU制御信号コードの格納はデータバス32から行う。2410はセレクタ回路であり、拡張機能指定信号1301の信号値に従い、拡張機能記憶装置1210から読み出されたALUの制御信号2210あるいはDEC1120の出力信号2120を選択し、最終的なALUの制御信号1410とする。1217はNOP指定信号である。1670および1680はALU1610への入力データを伝えるためのバス、1690はALUの演算結果を伝えるためのバスである。図25では、クロック信号PH1が1のときALU1610へのデータの入力が行われる。クロック信号PH2が1のとき拡張機能記憶装置1210からデータの出力が行われる。

【0162】以上の実施例では、全ての拡張機能記憶装置が同一のアドレス選択信号を用いる例を示した。各拡張機能記憶装置ごとにアドレス選択信号を変えることも可能である。また、アドレス選択信号はマイクロ命令を利用せずマイクロコントローラから与えることも可能である。各拡張機能記憶装置ごとに拡張機能指定信号を変えることも可能である。拡張機能記憶装置に格納するコードは、マイクロ命令で指定可能な（エンコードした状態の）演算機能指定コードとすることも可能である。

【0163】

【発明の効果】この装置を用いる結果、並列コンピュータに存在する元の命令からいくつかのプロセッサにおいて2次的な命令を生成し、異なるプロセッサでこれらの別々の命令を同時に実行することができる。

【図面の簡単な説明】

【図1】本発明になる、並列処理集積回路のアレイを用いたコンピュータシステムを示した図。

【図2】図1の理解を助ける波形図。

【図3】図1のアレイの1つのプロセッサの一例の具体

45

例を示すブロックダイアグラム。

【図4】図3のブロックダイアグラムに示すメイン処理ユニットのブロックダイアグラム。

【図5】図3のブロックダイアグラムに示すアドレス一致回路のブロックダイアグラム。

【図6】図3のブロックダイアグラムに示す外部メモリインターフェースユニットのブロックダイアグラム。

【図7】図5のブロックダイアグラムに示すアドレス一致回路の論理回路を示す図。

【図8】図5のブロックダイアグラムに示す命令修正回路の論理回路を示す図。

【図9】図7及び図8の回路の理解に役立つ波形を示す図。

【図10】図1のシステムの可能な動作モードの一つを図式的に示す図。

【図11】図1のシステムの別の可能な動作モードの一つを図式的に示す図。

【図12】従来の単一の命令の流れによって駆動される並列処理集積回路のアレーを用いたコンピュータシステムの図。

【図13】本発明を適用したMIMDアーキテクチャの並列プロセッサシステムを概略的に示すダイアグラム。

【図14】図13のシステムの動作モードの一つを図式的に示すダイアグラム。

【図15】本発明の第1の実施例と第2の実施例の原理

46

の違いを示す図。

【図16】本発明の第2の実施例を示す図。

【図17】図16の命令決定回路の詳細を示す図。

【図18】図17のセレクトの詳細を示す図。

【図19】図16の装置に用いる算術論理演算回路の図。

【図20】図16の装置における、複数のプロセッサグループを示す図。

【図21】本発明による他の並列プロセッサの例を示す図。

【図22】図21の装置におけるプロセッサの詳細を示す図。

【図23】図21の装置に用いるマイクロ命令のフォーマットを示す図。

【図24】図21の装置における、マイクロ命令と各プロセッサグループの演算との関係を示す図。

【図25】本発明による、さらに他の実施例を示す図。

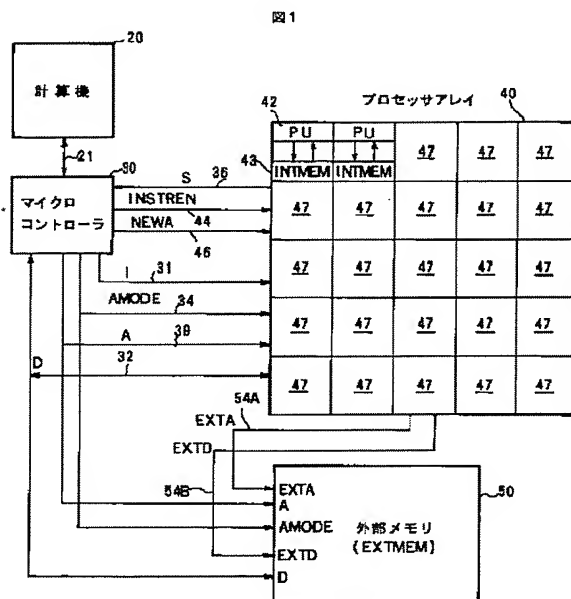
【図26】図5のプロセッサIDレジスタ(84)の動作のタイムチャート。

20 【図27】図5のアドレス一致マスクレジスタの動作のタイムチャート。

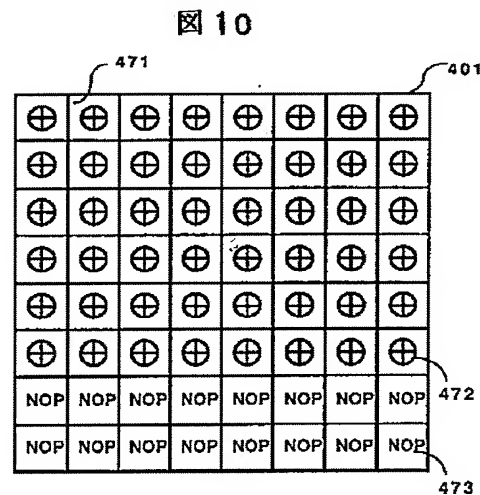
【符号の説明】

47…プロセッサ、31…命令バス、32…データバス、34…アドレスモード信号バス、36…ステータスフラグバス、39…アドレスバス

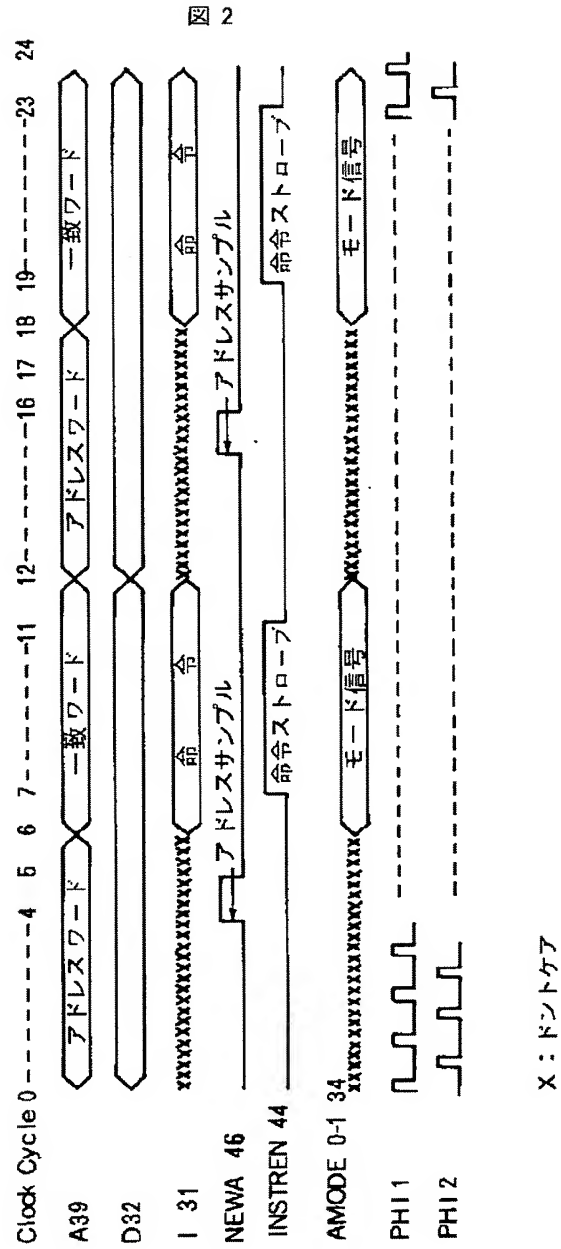
【図1】



【図10】



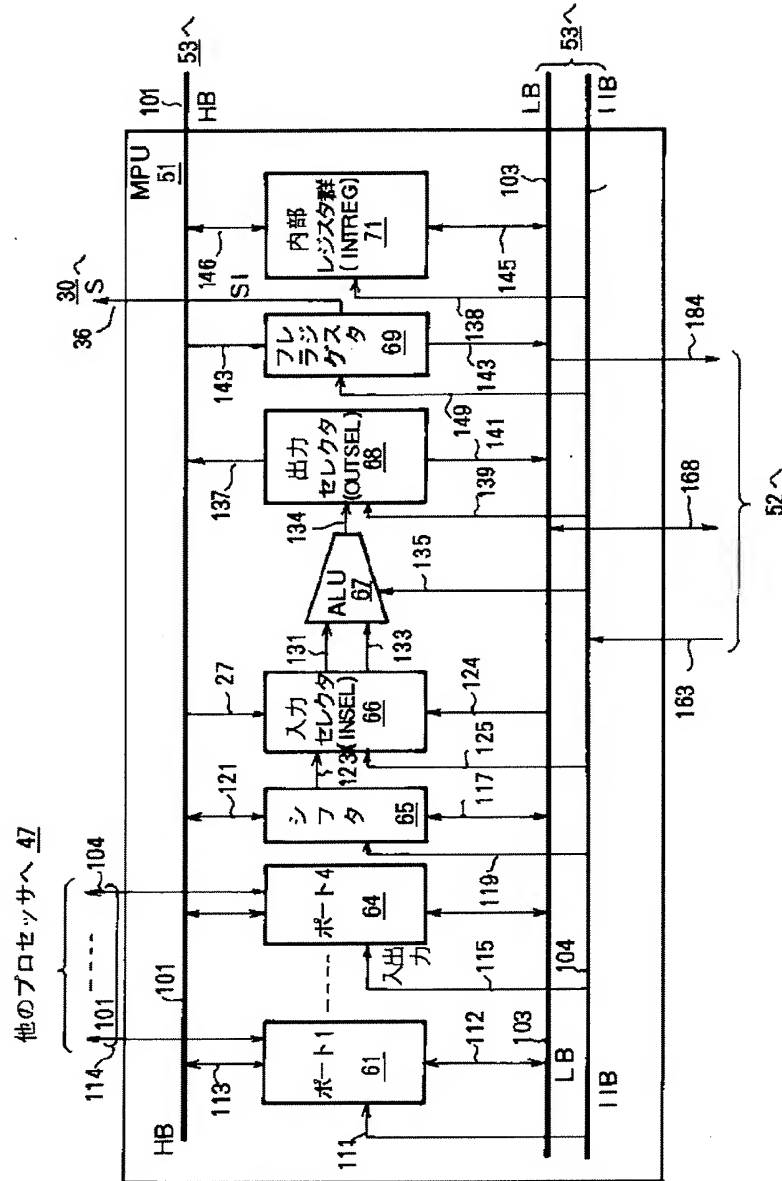
【図2】





【図4】

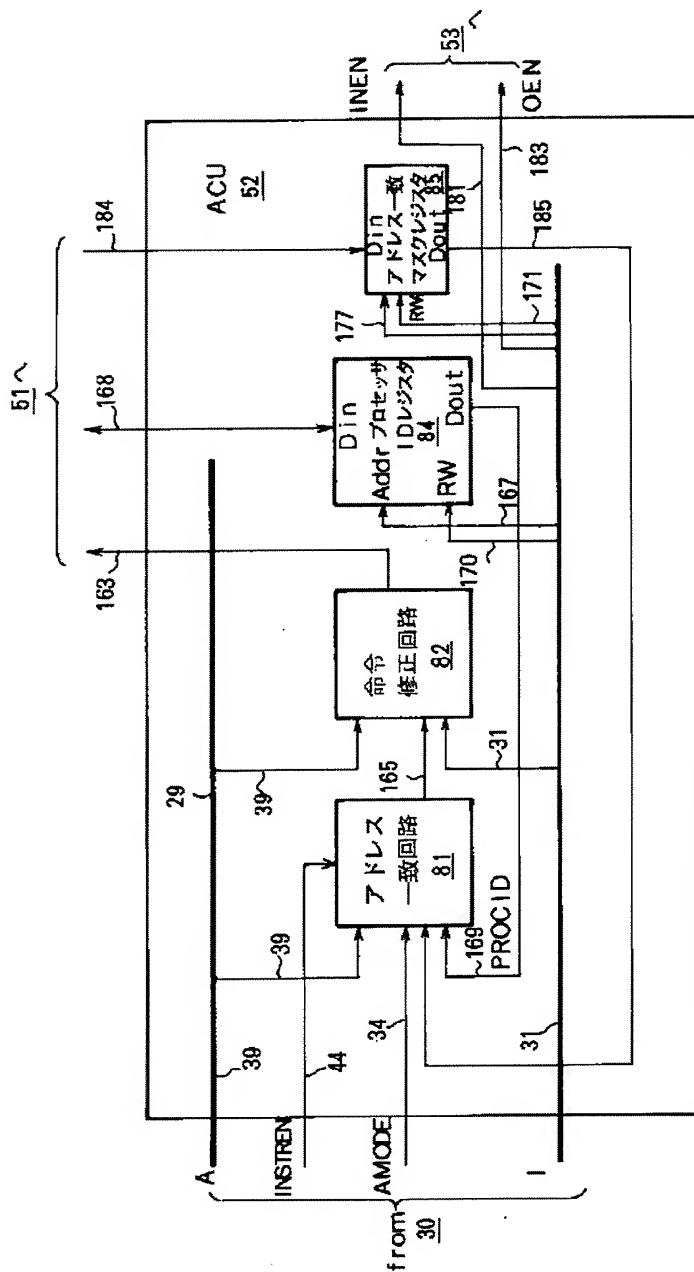
図 4





【図5】

図 6

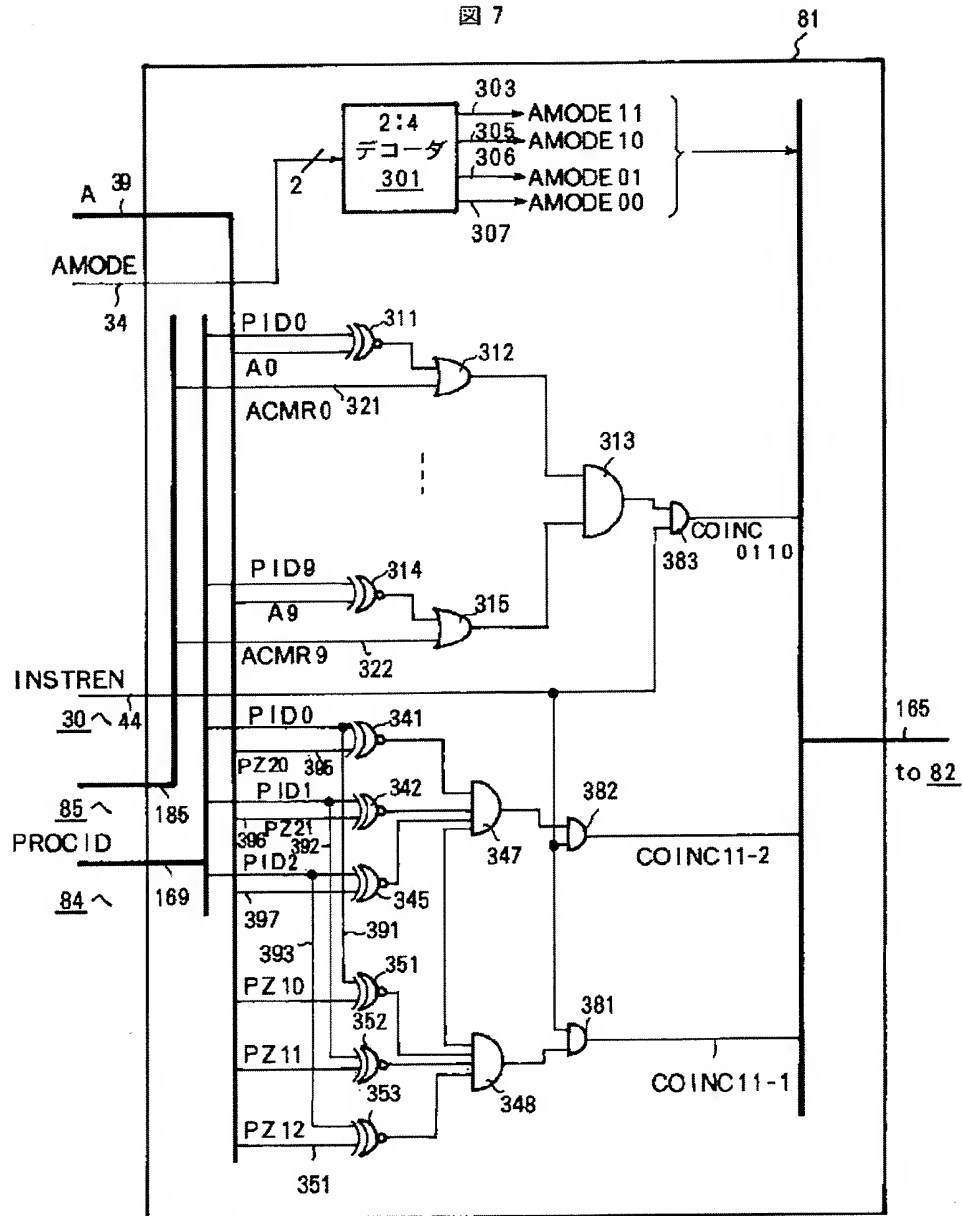


6



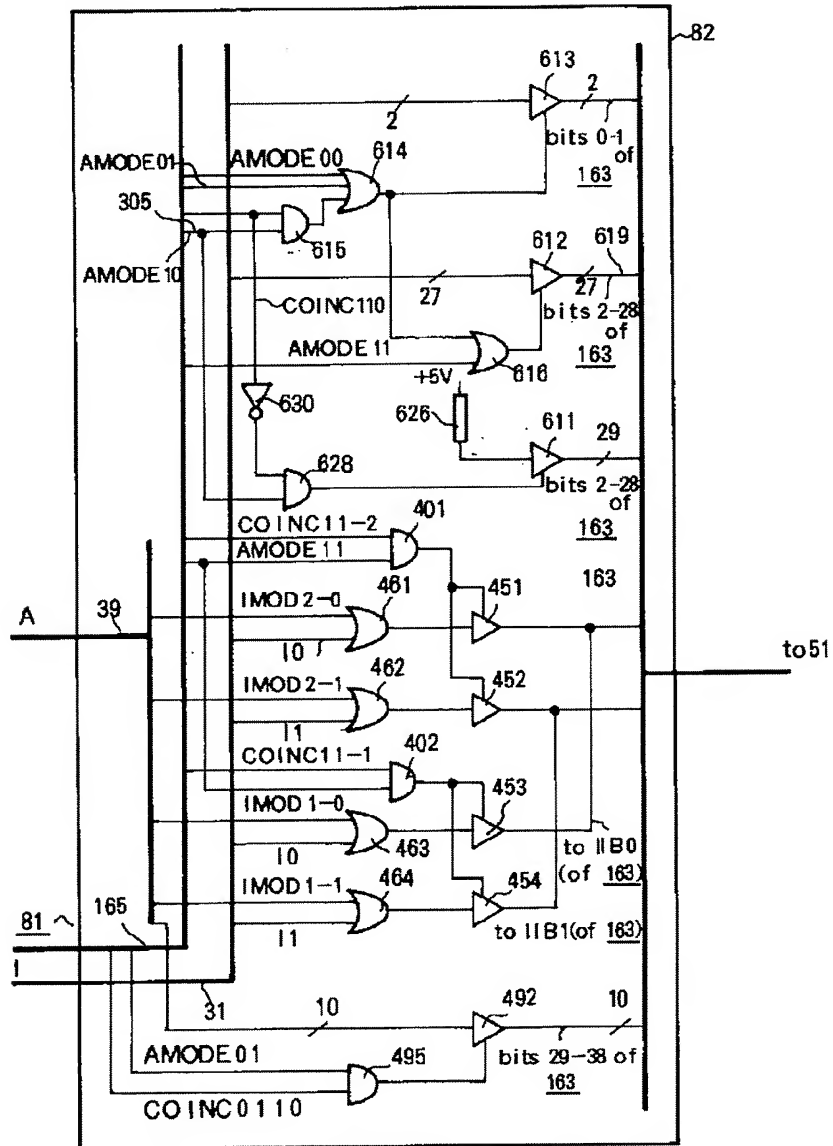
【図7】

図 7



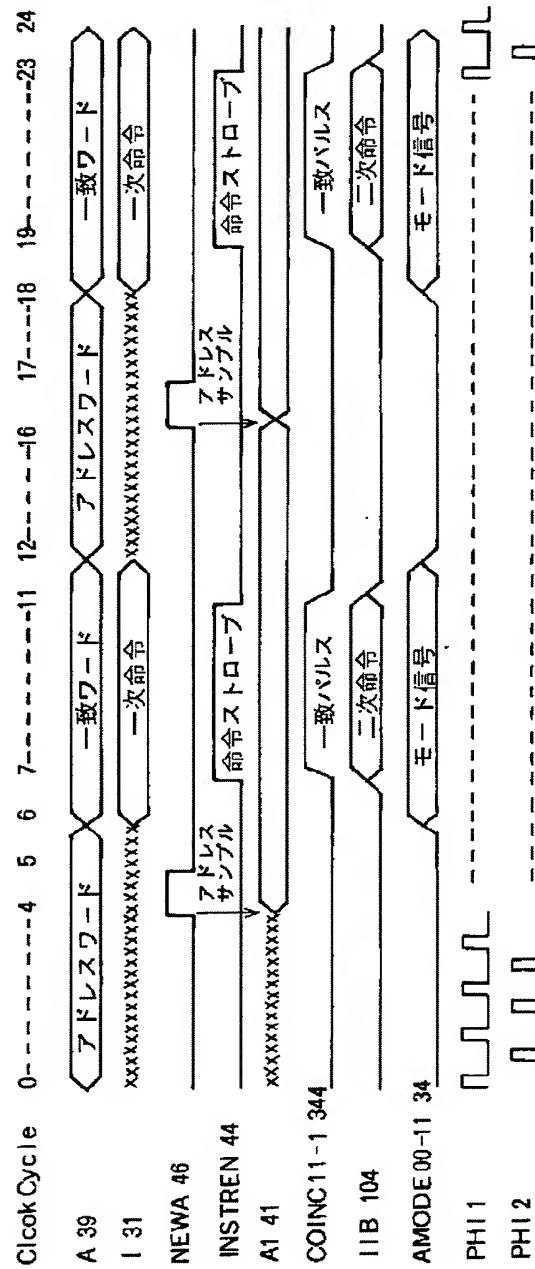
【図8】

図 8



【図9】

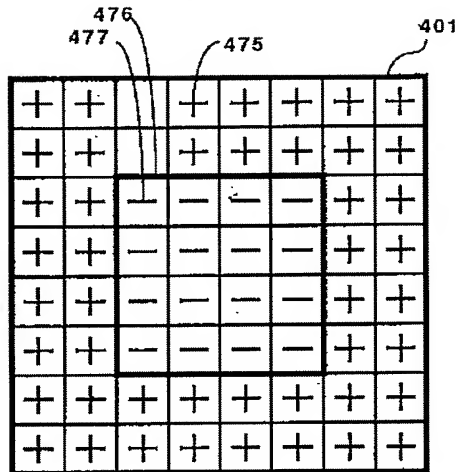
図9



X : ドントケア

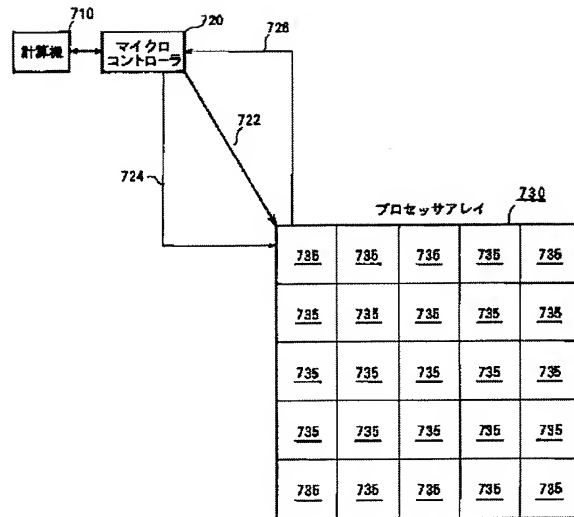
【図11】

図11



【図12】

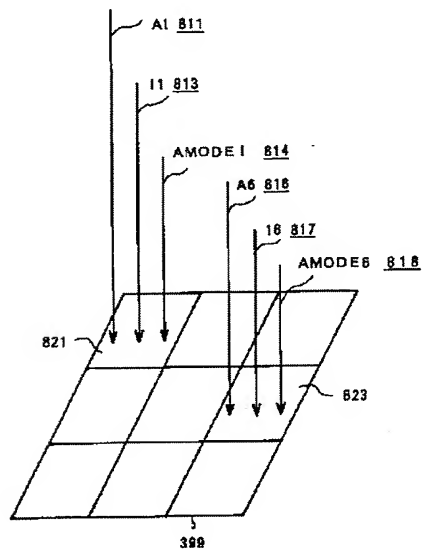
図12



【図13】

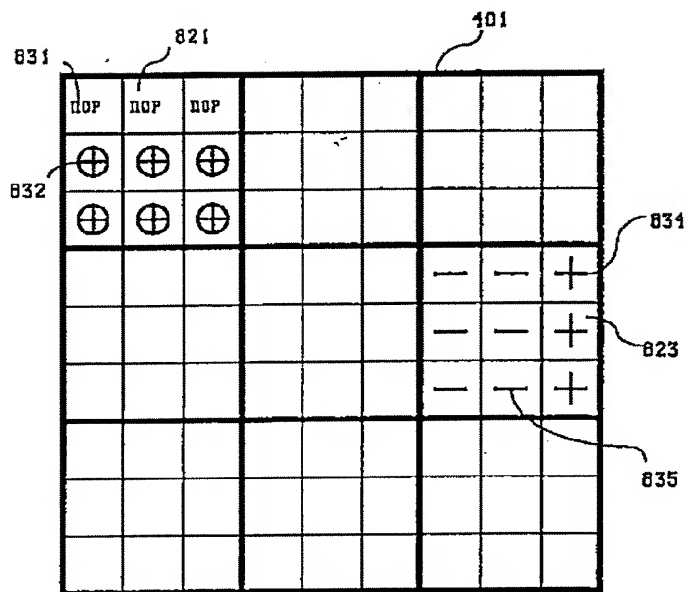
図13

マイクロコントローラ 03 から



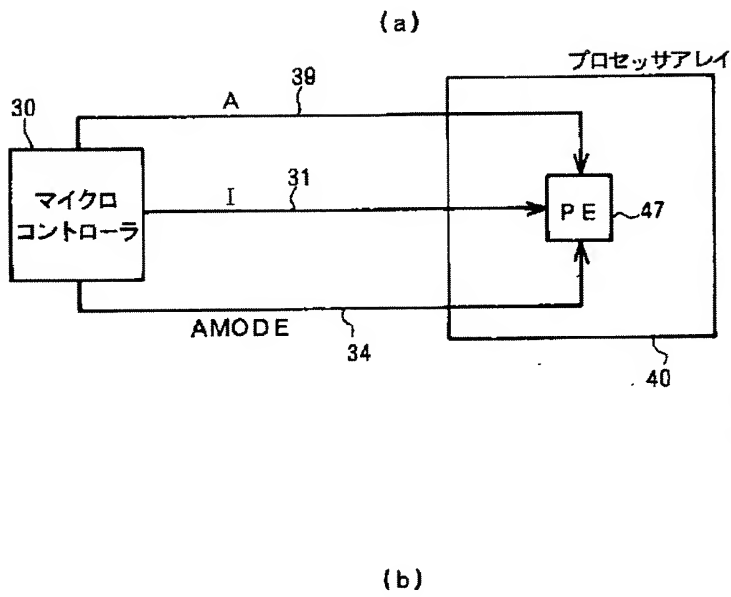
【図14】

図14



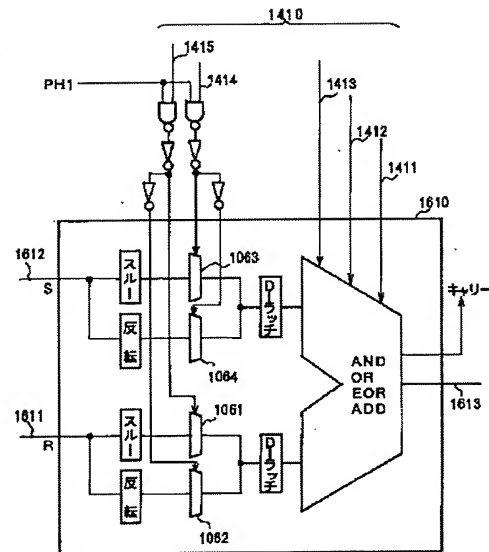
【図15】

図15



【図19】

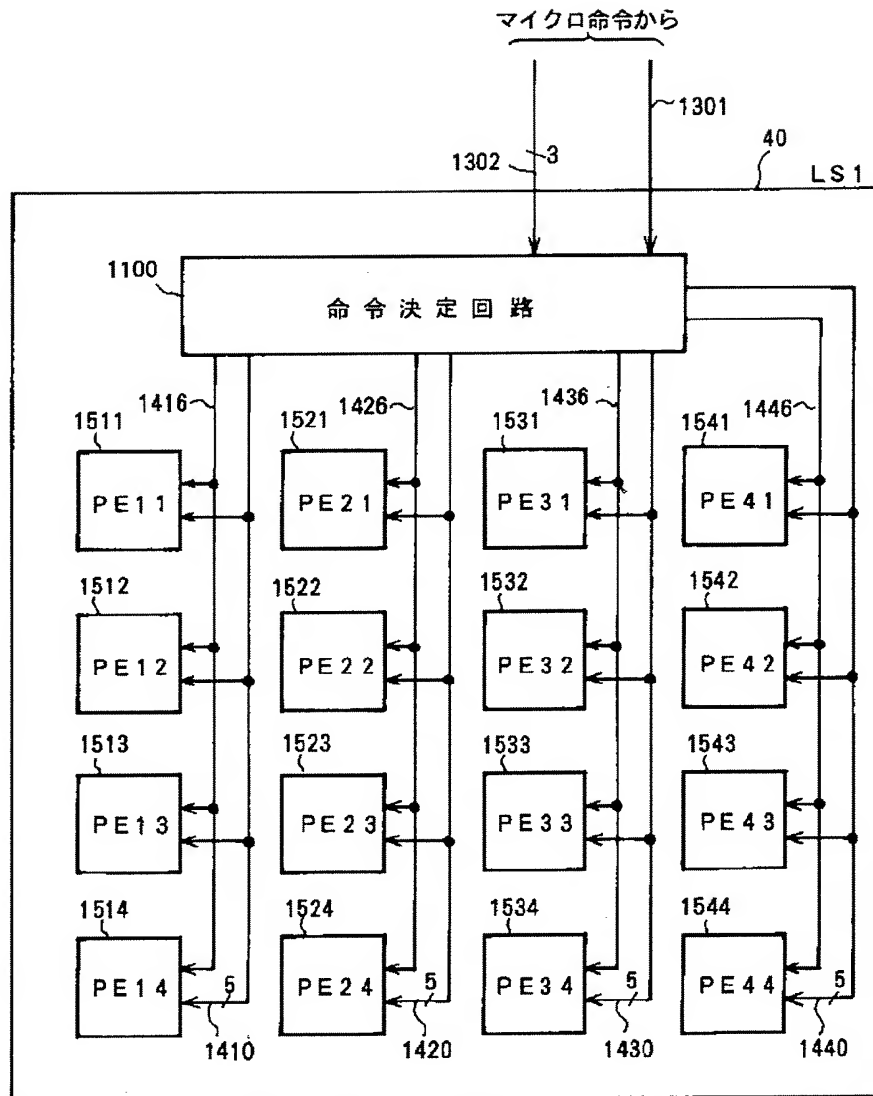
図19





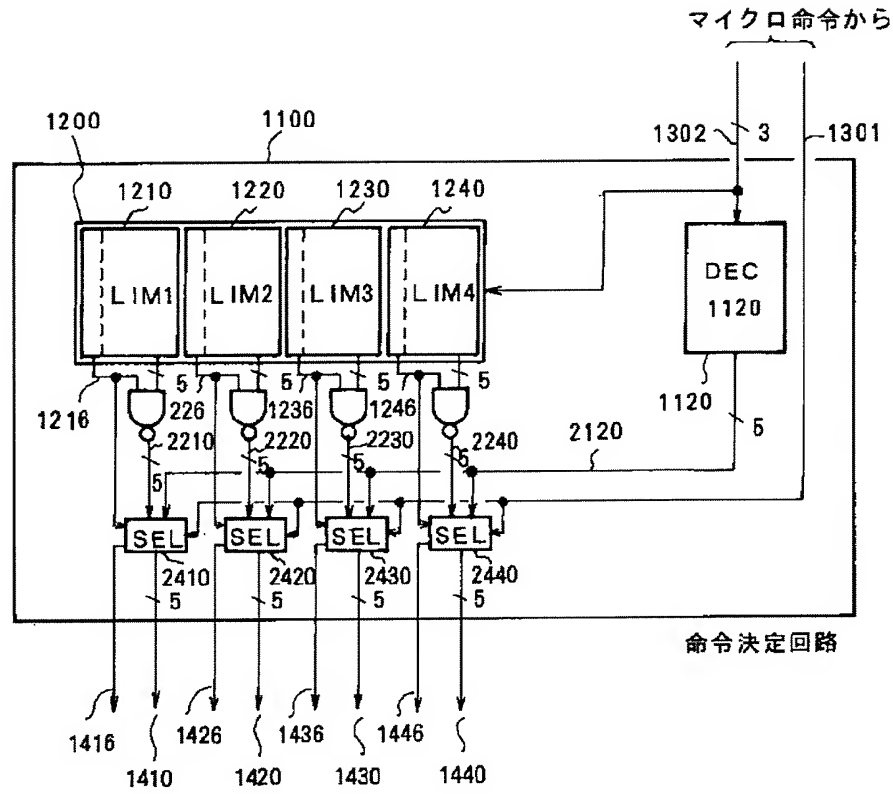
【図16】

図16



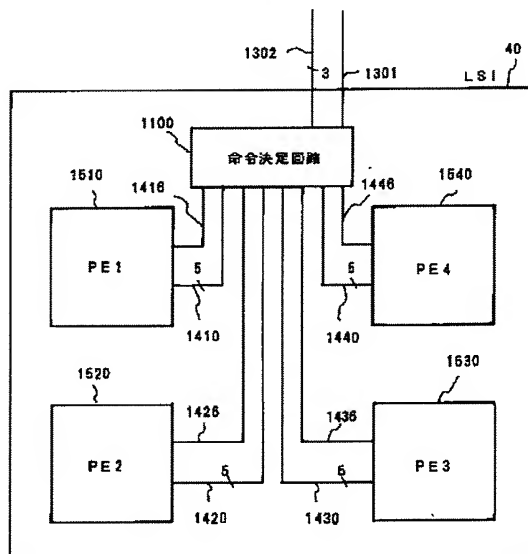
【図17】

図17



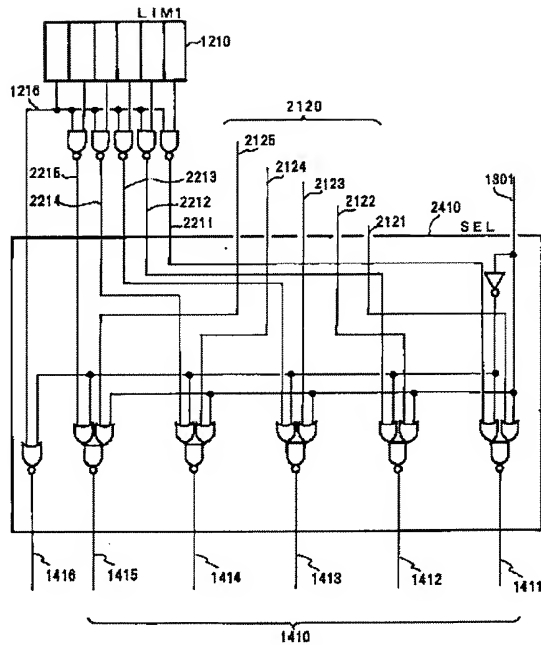
【図21】

図21



【図18】

図18

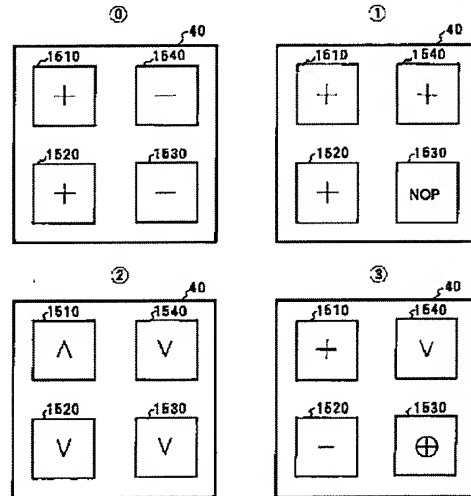


【図20】

図20

	LIM1 1210	LIM2 1220	LIM3 1230	LIM4 1240
⑦				
⑥				
⑤	1 1 1 1 0	1 1 0 1 1 0	1 1 1 0 1 0	1 1 1 0 0 0
④	1 1 1 0 1 1	1 1 1 0 0 0	1 1 1 0 0 0	1 1 1 0 0 0
③	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0
②	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0	1 1 1 1 1 0
①				
⑦				

\* : ドントケア



【図23】

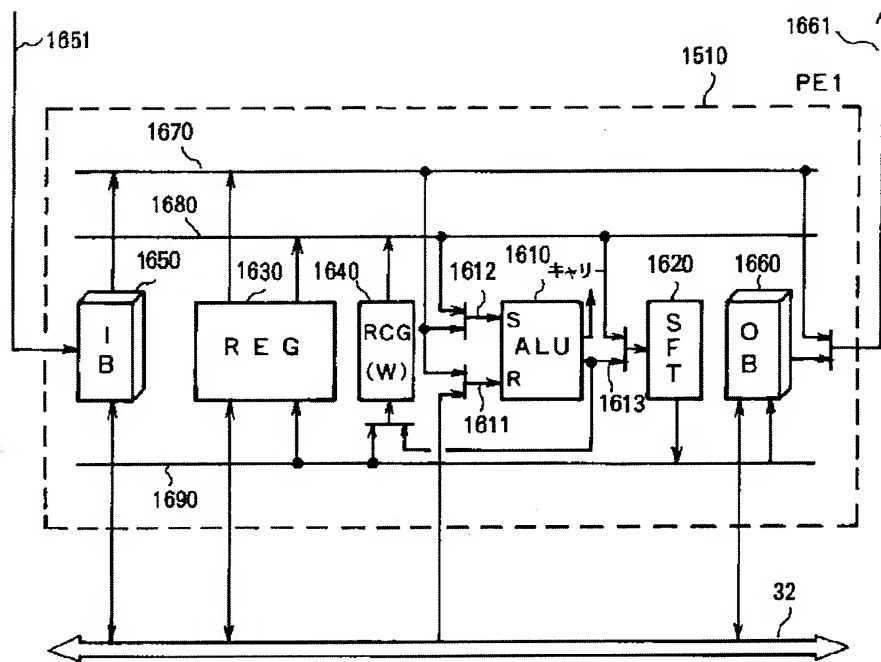
図23

フィールド1		フィールド2		フィールド3		フィールド4		フィールド5		
バス-A 制御		バス-B 制御		バス-C 制御		バス-D 制御		演算制御		
S 指定	D 指定	S 指定	D 指定	S 指定	D 指定	S 指定	D 指定	ALU 入力	ALU 機能	ALU 出力

注) S:Source, D:Destination

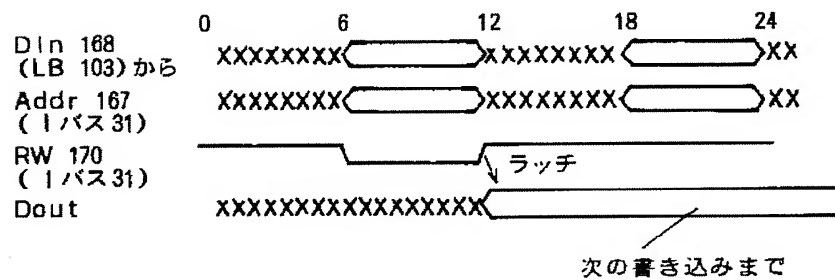
【図22】

図 22



【図26】

図 26



【図24】

図 24

